2025 IEEE 21st International Conference on Automation Science and Engineering (CASE)
August 17-21, 2025. Los Angeles, CA, US

2025 IEEE 21st International Conference on Automation Science and Engineering (CASE) | 979-8-3315-2246-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/CASE58245.2025.11163896

Coverage Path Planning for Ultrasonic Non-Destructive Inspection

Jonas Beachy[1] and Xu Chen[1]

***Abstract*— Non-destructive inspection of composite materials is a critical process in industries such as aerospace and construction, where structural integrity must be verified to prevent catastrophic failures. Ultrasonic testing is a widely used NDI method for detecting defects in composite parts, but its effectiveness depends on maintaining a water column between the probe and the part surface. Traditional automated coverage path planning (CPP) approaches fail to ensure full coverage, particularly along the edges of complex geometries where the loss of the water column can compromise inspection reliability. This paper presents a novel CPP method for robotic NDI that enhances inspection coverage by fitting the edges with parametric Bézier curves to ensure inspection around the edges before constructing the remainder of the path. The method then refines the path using a modified potential field algorithm and a redundant coverage elimination algorithm. The result of the proposed method is a largely automated, efficient NDI process while maintaining a high quality of inspection.**

I. INTRODUCTION

Inspection of manufactured parts is an important task in industries like aerospace and construction, where it is critical to verify the part's ability to withstand the loads for which it was designed. These tasks become even more important when inspecting materials that have catastrophic failure modes with little external warning. One such material is carbon fiber composites, which are stronger than steel and lighter than aluminum, making them increasingly popular in aerospace and automotive applications for their exceptional strength-to-weight ratio.

Non-destructive inspection (NDI) is a broad category encompassing a variety of methods used for inspection; eddy-current testing, visual inspection, and radiographic testing are all examples of NDI techniques. A common inspection method for composite materials is ultrasonic testing (UT) with a phased array ultrasonic probe. In UT, the probe emits high-frequency acoustic waves and detects reflections, or echoes, as the wave propagates through the part. Sub-surface defects can be determined and visualized from the time-of-flight of the echoes. UT can detect common defects found in composite, such as delamination, wrinkles, and porosity. The resolution needed to detect flaws in composite parts requires ultrasonic frequencies in the range of 3-10 MHz. At such high frequencies, ultrasound waves cannot propagate through the sparse atmosphere, instead relying on an intermediary material, or couplant, to transfer the wave to the part. Typically, water is chosen as the couplant due to its availability and non-reactive properties. To produce high-quality images, a consistent column of water between the probe and part is essential to ensure that a strong and reliable wave reaches the part, illustrated in Figure 1.

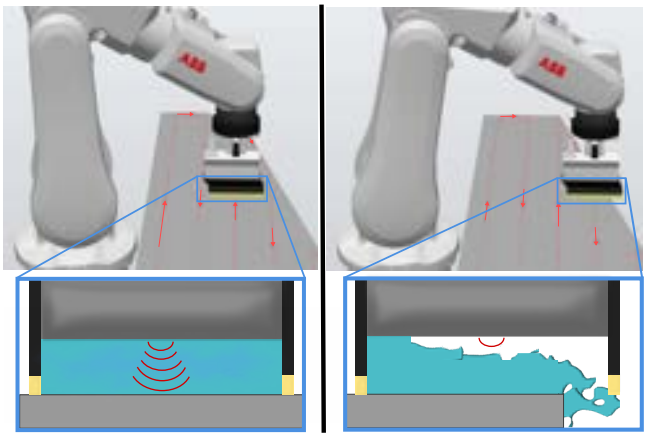


Fig. 1: Illustration of water-column loss near edges. A hose attached to the end-effector continuously supplies water from an external tank (Not shown).

While some ultrasonic NDI is done by hand, to efficiently inspect a surface in large-scale manufacturing the process must be automated, such as integrating the probe to the end-effector of a 6 degree-of-freedom robotic arm manipulator. Given that sub-millimeter positional accuracy in modern industrial robot arms is feasible, an offline-generated path can be given to the robot to carry out the inspection with sufficient accuracy. To automate the NDI effectively, the task becomes planning a robot path that ensures the entire surface has been scanned under the probe while respecting NDI-specific constraints such as probe orientation and maintaining a water column.

The above challenge is classified as a coverage path planning (CPP) problem (see Section II) and has been approached in literature and with commercial products. In these previous works, path planning is performed by overlaying a simple raster pattern over the surface [1], or by selecting an edge and then moving parallel with the surface by a probe-width [2], [3]. Neither method will provide the needed full coverage for aerospace parts because of excessive water loss around the edges of non-rectangular surfaces, which causes the loss of UT data, shown in Figure 1. The edges are critical regions to scan as localized stresses make them particularly susceptible to flaws [4].

To address the current limitations of CPP, we propose

This work was supported by the Boeing Advanced Research Collaboration at UW. The authors acknowledge and thank Jill Bingham, Andrew Na, Bill Moon, and Shannon Dong for their guidance in the foundational work underlying this study.

[1]Authors are with the Department of Mechanical Engineering, University of Washington, Seattle, WA 98195, USA jbeachy@uw.edu, chx@uw.edu

a novel CPP workflow for non-destructive inspection that provides two main contributions:

1) Generating a coverage path that ensures the edges are inspected properly by fitting parametric Bézier curves.
2) Refining the path by reducing redundant coverage and using a modified potential field algorithm.

The combination of these two contributions with other processes in the paper results in a robust system that can automate the inspection of a wide variety of composite parts. The rest of the paper is organized as follows. Section II provides a brief overview of related work, and the proposed method is shown in Section III and IV. We discuss the results in Section V and conclude the work in Section VI.

II. Related Works

Coverage path planning is a subfield of robotic path planning concerned with ensuring complete coverage of a target area while avoiding obstacles, obeying task-specific constraints, and maintaining path efficiency. It is a robust area of robotics research, highly suited for inspection tasks, and as seen in the seminal survey paper [5], has applications in domains as varied as floor-vacuum robots to undersea structure inspection. Much of the work in CPP is done assuming a two-dimensional (2D) workspace and relies on structures such as grid-based maps [6]. These structures are difficult to generalize into 3D, often requiring problem-specific adaptations. Agricultural plowing [7], automotive spray painting [8], and underwater structure inspection [9] are examples that uniquely address 3D CPP challenges.

Several studies have explored CPP in the context of manufacturing focused non-destructive inspection. For eddy-current testing, [1] proposes a method for decomposing complex parts into simple surfaces and overlays a raster pattern path, assuming all decomposed surfaces are planar. Specifically for UT, [2] introduces a method that propagates a path from a selected mesh edge across a part at fixed curvilinear intervals. Alternatively, [10] computes streamline-based coverage passes using mesh face normals and then connects them in a path by solving a traveling salesman problem. Our method aims to build on both works by prioritizing high-quality data collection around the edges and forming efficient scanning paths.

III. Coverage Path Planning Generation

The proposed method used to construct an initial coverage path plan is detailed in this section. While not the focus of this paper, pre-processing methods are applied to scan and segment the part, enabling the inspection of a wide variety of parts (Section A). Then the principal scanning direction, corners, and edges of the surfaces are determined for each segment (Sections B and C). Finally, the edges are fit with a parametric Bézier curve and interpolated to guarantee coverage (Sections D and E).

A. Optical Scanning and Segmentation

When handling composite parts, there may be significant deviations between the CAD and the part stemming from distortion due to gravity or fixturing. Because of this, planning an inspection path based on the CAD will lead to collisions or water-column loss. To avoid these outcomes, we image the part with a laser or structured light scanning system. This gives an as-manufactured, as-fixtured point cloud model of the part the robot can use to path plan within the error tolerance of the imaging system. Then we transform the point cloud into the robot's base coordinate system using a standard hand-eye calibration method [11]. Finally, the point cloud is formed into a triangulated mesh using the ball-pivoting method [12]. The result of this process is an stl-like surface mesh with vertices and face normals that can be used for path planning while reflecting the true shape of the part. Since UT requires scanning from only one side, the target surface should be oriented upward during the inspection process to ensure unobstructed probe access.

The goal of the segmentation step is to decompose the surface mesh into segments such that each segment is a surface well-posed for our proposed method. This requires each surface to have no interior holes, and the projection onto the plane formed by the principal and secondary axis (see Sec III-C) be a bijection. The bijection requirement ensures segments remain unambiguous under the proposed method. For complex parts with multiple distinct faces, methods like the watershed algorithm or face normal clustering can be used as in [1]. However, by the nature of the composite lay-up manufacturing process, most parts are already well-posed for inspection and rarely need segmentation. With this in mind, the segmentation in this work is an extension into 3D of the conventional CPP Morse decomposition algorithm [13]. Morse decomposition sweeps across the part along the principal axis, identifying critical points where the continuity of the sweep line changes due to interior holes or part geometry. At the critical points, new segments are formed. It is not a requirement for the segments to be convex, but highly irregular shapes will be segmented into simpler forms.

B. Principal Scanning Direction

From the pre-processing, we are given a collection of mesh segments that each need to be inspected. Thus, a coverage

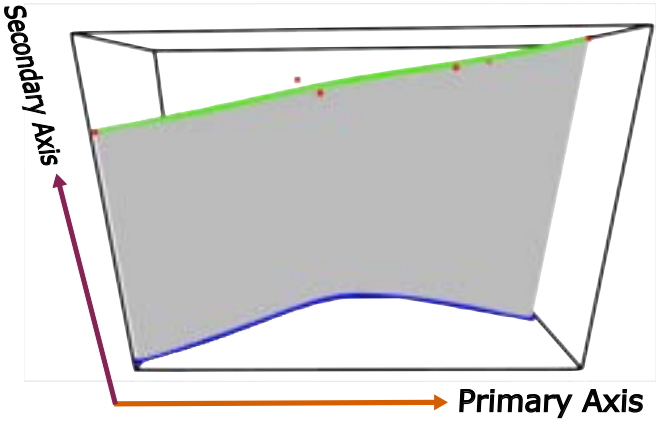


Fig. 2: Airfoil surface with principal component bounding box and Bézier curve fit edges along the primary axis. Control points of the top curve are shown in red.

path is planned for each of the segments individually using the following procedure.

We choose the following reference frame: The scan direction is the direction in which the ultrasonic probe travels along a single scanning pass, while the index direction is the direction the probe moves between passes. The index direction is typically close to perpendicular to the scan direction in a traditional raster pattern. A visualization of the two directions is shown in Fig. 2. We apply principal component analysis (PCA) to determine these directions for the segment surface.

PCA is a widely used tool in data analysis for decomposing high-dimensional data into an ordered orthogonal basis of principal axes. We first compute the covariance matrix of the mean-centered data and then perform the PCA using eigendecomposition. From all N mesh vertices, we can construct a N×3 matrix $P = \{p_1, \ldots, p_N\}$, where each $p_i = [x_i, y_i, z_i]$ is a vertex in 3D space. The mean-centered data, $\bar{P}$ is written as

$$\bar{P} = P - \frac{1}{N}\sum_{i=1}^{N} p_i, \tag{1}$$

which can then be used to find the covariance matrix

$$C = \frac{1}{N}\bar{P}^T\bar{P}. \tag{2}$$

Because $C \in \mathbb{R}^{3x3}$ is a positive semi-definite matrix by construction, we are guaranteed eigenvalues $\lambda_i > 0$. Thus, the eigendecomposition of C yields the set of ordered eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$ with corresponding eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$. Each eigenvector is a unit column vector of length 3 describing a principal direction in 3D space. Eigenvector $\mathbf{v_1}$ is the direction that captures the most variance in the segment mesh and therefore is set as the primary scan axis. Similarly, $\mathbf{v}_2$ and $\mathbf{v}_3$ are the secondary and tertiary scan axis directions, each capturing the most variation in the data while maintaining orthogonality to preceding directions. The bounding box formed along the scan directions can also be seen in Fig. 2. Since the greatest variance in the mesh is captured by the primary scan axis, we align the scan direction with it. Longer passes and fewer index-direction movements minimize unnecessary repositioning and improve overall efficiency in the scan.

C. Corner and Edge Identification

To find the edges that align closely with the primary scanning axis, the corners that delineate one edge from another must be found. We achieve this by extracting the edge points from the mesh. On a triangle mesh, the edge is made of line segments between edge vertices. Line segments only part of one triangle make up the edge, while all other interior line segments will be part of two triangles. The edge segments can be found by iterating through the mesh triangle data and tracking the occurrence of each line segment. From there, we collect the edge vertices into the set of edge points, P_E.

To identify corner points, we project the set of edge vertices into 2D to emphasize local angle changes. To minimize the information loss, P_E is projected on the plane formed by the primary and secondary scan axes, calculated with

$$\tilde{P}_E = P_E\left[\mathbf{v}_1\ \mathbf{v_2}\right]. \tag{3}$$

In 2D, the $\tilde{P}_E$ points are first ordered in a clockwise fashion based on the angle to an interior point. Then for each $\tilde{P}_{E_i}$, the dot product between the preceding and following line segment is calculated. A dot product absolute value near 0 signals the line segments are close to perpendicular and therefore $\tilde{P}_{E_i}$ is a corner point candidate. The edge points with the lowest dot product values are saved as the corner points. In cases with noisy edges, too few edge points, or too many edge points, we prompt an operator for manual corner point selection.

With the corner points selected, distinct edges are defined as the groups of edge points between corner points. We select the two edge groups that have the longest curvilinear distance while aligning with the principal axis as the primary boundary edges E_1, E_2, and they are the edges that must be followed closely to ensure efficiency and ultrasonic scan quality.

D. Edge fitting

To fit boundary edges E_1, E_2, we repurpose the mathematical tool of a Bézier curve here. Widely used in computer graphics, Bézier curves are known in the robotics community due to their ability to generate smooth trajectories that are easy for robots to follow [14]. Beyond smoothness, a Bézier curve is used for two reasons:

- The fitting acts as a low-pass filter, removing possible residual noise in boundary edges from imaging.
- The parametric nature of the curve lends itself to rapid sampling of evenly-spaced points along its curve.

While other fitting methods such as cubic spline interpolation can also fit smooth curves, the parametric nature of Bézier curves is valuable for efficiently generating paths as seen in Algorithm 1. The Bézier curve takes the form

$$B(t) = \sum_{i=0}^{n}\binom{n}{i}(1-t)^{n-1}(t)^i\mathbf{P}_i \quad 0 \leq t \leq 1, \tag{4}$$

where n is the order of the curve and $\mathbf{P}_i \in \mathbb{R}^3$ are the $n+1$ control points. The location of the control points determines the shape of the curve as the parametric variable t moves from 0 to 1. To fit a boundary edge, the control points must be placed in such a way that the resulting Bézier curve matches the true edge curve quite closely. The corner points E_{c0}, E_{c1} are set to be P_0 and P_{n+1} because they are defined as start and end points of the curve when $t = 0$ and $t = 1$. We find an approximation of the optimal location of the remaining control points by minimizing the distance between the Bézier curve and the boundary edge at m different locations. We take t_m as the set of evenly spaced points on the t axis $\{t_j = \frac{j}{m} | j \in \{0, \ldots, m\}$. Then the corresponding vertices from E is the set $\{E_j | \frac{s(E_j)}{S} \approx t \in t_j\}$ where S is the total curvilinear distance of E and $s(E_j)$ is the cumulative curvilinear distance from E_{c0} to E_j. Minimizing

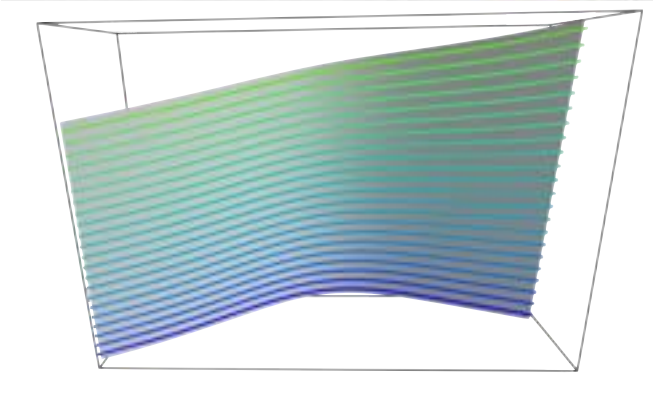


Fig. 3: Sampled scan passes for the airfoil; $N_{pass} = 22$.

the distance can be formulated as solving the optimization problem for the set of control points P:

$$\min_{P} J = \sum_{j=0}^{m} \|B(t_j) - E_j\|^2. \tag{5}$$

This problem can be constructed into a disciplined convex program and solved with any least-squares solver. The result of the optimization is the control point set that forms a parametric Bézier curve. When applied to E_1 and E_2, the resulting control points, P_{E1}, P_{E2}, form curves fitting the boundary point set, and are the basis upon which the complete coverage path is constructed. An example of the edge fitting is shown in Fig. 2 and the Bézier control points of the top curve are shown in red.

E. Path Construction

The coverage path is made up of passes joined together such that they form a raster-like pattern; The passes model the on-surface location of the robot's tool center point (TCP), which is assumed to be centered under the width and breadth of the ultrasonic probe. During inspection, the probe scans one probe-width of surface area as the robot moves along the inspection passes. The probe-width is oriented perpendicular to the pass, generally aligning with the secondary axis. Additionally, each pass is connected to the next with a motion in the secondary axis direction. Thus, the number of passes needed to completely scan the surface, N_{pass} is found by dividing the maximum curvilinear width of the mesh along the secondary axis by the functional scan width of the probe and rounding up to the next largest integer with a ceiling function. We find the maximum curvilinear width by slicing the mesh with planes normal to the secondary axis and summing the distances along the intersection curve between the mesh and the slicing plane. N_{pass}, then, is the number of intermediate passes needed between the E_1 and E_2.

The proposed procedure for generating the pass points is shown in Algorithm 1. First, the edge control points sets P_{E1} and P_{E2} are shifted toward the center of the mesh by a half-probe width along the secondary axis. This allows the probe to follow the edge while maintaining a water column. The remaining intermediate passes are found by taking convex combinations of the shifted edge control points, resulting in $N_{pass} - 2$ sets of control points whose curve shape smoothly interpolates between the two curves. Taking advantage of the parametric formulation of Bézier curves, points evenly spaced along the curve can be found very efficiently with an implementation of Newton's method (*ArcLength()* in 1). Using the arc-length formula for parametric equations and the previous pass point, Newton's method quickly converges to the t_{new} value exactly distance d from t_{old} along the curve. The sampled point in 3D space is found by evaluating the Bézier curve at t_{new}. The distance d can vary based on the desired point density but should be no greater than the probe breadth to ensure the path refinement step goes smoothly.

Because the edges are interpolated to form the passes, there is no guarantee that the intermediate passes remain on the mesh surface. To get the on-surface pass point, we ray-trace along the tertiary axis to find the mesh intersection. The on-surface passes are shown in Fig. 3. We find the pose of the on-surface point is by the average vector to the preceding and following point (x-axis), the average surface normal of the intersected and all neighboring triangles (z-axis), and the x-z cross product (y-axis).

Algorithm 1 Interpolation and rapid sampling of scan passes.

Input: Control Point sets: $\mathbf{P}_{E1}, \mathbf{P}_{E2}$, number of passes: N_{pass}, desired distance: d
Output: N_{pass} Coverage passes **S**=$\{s_1, s_2, \ldots, s_N\}$ sampled with regular intervals of length d along the curve

$C_1, C_N \leftarrow \mathbf{P}_{E1}, \mathbf{P}_{E2}$ shifted inward $\frac{1}{2}$ probe-width
for $i \leftarrow 1$ to N_{pass} **do**
  $\lambda \leftarrow \frac{i-1}{N-1}$
  $\mathbf{C}_i \leftarrow (1-\lambda)\mathbf{C}_1 + \lambda \mathbf{C}_N$
  $s_i = [\ ]$
  $t_{old} \leftarrow 0$
  while $t_{new} \leq 0$ **do**
    $t_{new} \leftarrow$ *ArcLength*(t_{old}, d)
    Append *BézierValue*(C_i, t_{new}) to s_i
    $t_{old} \leftarrow t_{new}$
  end while
end for
return **S**=$\{s_1, s_2, \ldots, s_N\}$

IV. Path Refinement and Coverage Guarantee

In cases where the PCA of the segment is degenerate, defined as situations where the variance is roughly spread evenly throughout the three principal components ($\lambda_1 \approx \lambda_2 \approx \lambda_3$), gaps may exist in the initial coverage due to the surface variability and require the path to be refined. To address this, we calculate a scan-map showing what areas are covered. The scan map is built by forming a k-d tree from all the vertices in the segment mesh. The k-d tree allows for computationally efficient lookup of nearby points due to

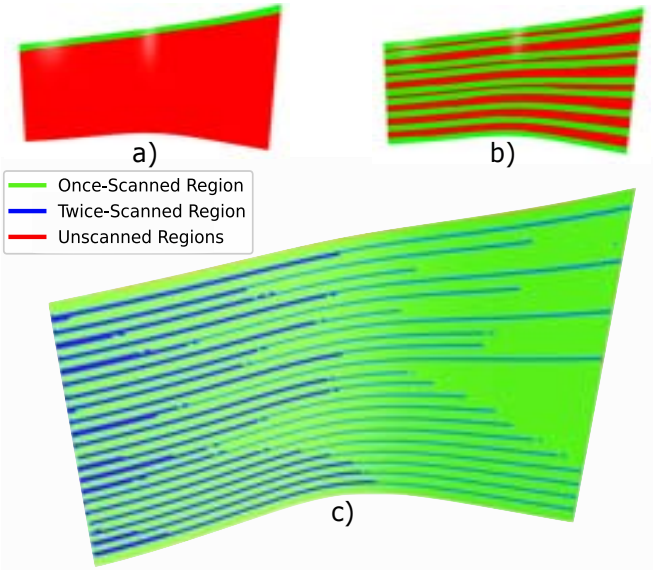
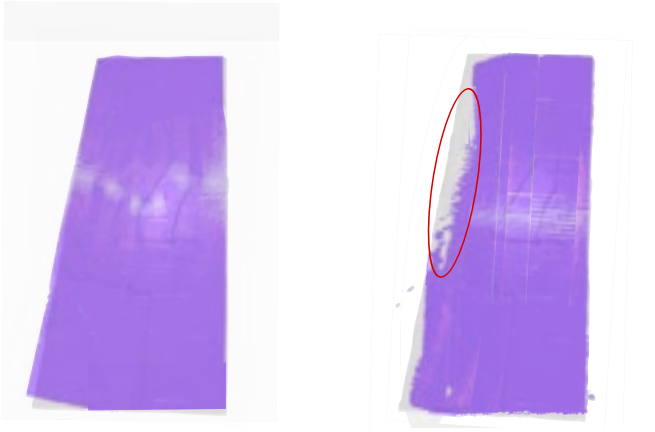


Fig. 4: Scan map for the airfoil after 1 pass (a), 9 passes (b), and N_{pass} passes.



(a) NDI data collected with our proposed CPP technique.

(b) NDI data collected with edge propagation.

Fig. 5: NDI data collected to demonstrate the difference in CPP methods.

its spatial structure. With the pose orientation from the on-surface pass points, the vertices inspected by the probe at each pass point are found and stored. After repeating for the entire set of passes, we create the scan-map by coloring the vertices based on the number of passes it is scanned by. Green signifies regions scanned only once, blue regions are scanned multiple times, and red regions are unscanned. A scan map is shown in Fig. 4.

The first path refinement process is a modified potential field algorithm. We model the unscanned vertices as vector field sinks, drawing points in with an attractive force. The magnitude of the attraction decays exponentially by distance so nearer points have much more influence. For each pass point p, all points in neighborhood $\mathcal{N}$ are evaluated and a correction vector δ is calculated by

$$\delta = \sum_{v_i \in \mathcal{N}_r} \omega_i (v_i - p), \quad \omega_i = \exp(-\frac{1}{\sigma}\|v_i - p\|), \qquad (6)$$

where $\mathcal{N}_r$ is the set of all unscanned vertices in $\mathcal{N}$, and σ parameter controls the decay rate. From our testing, $\sigma = 20$ produced good results when working in millimeters. In order to maintain the smoothness of the paths, total displacement is limited to 3 mm for each point along the normalized delta vector while motion of the edge passes is heavily constrained to maintain edge fit. The update is applied to all points (if no red vertices in $\mathcal{N}$ then they can be skipped), and the scan-map is recalculated for the updates passes. This process iterates until the red areas are covered. This can be seen in the difference between the Initial scan-map column and the Refined scan-map column of Fig. 6.

The second path refinement process focuses on the elimination of redundant pass points. We define a pass point as redundant when the area scanned by the point would remain completely covered by other passes if it were removed. While N_{pass} is necessary for coverage at the widest curvilinear length of the mesh along the secondary axis, there can exist redundancy in the path depending on the shape. A mesh that is wide at one end of the principal axis and narrow at the other end will contain many redundant points, shown as orange points on the Nose cone in Fig. 6. The redundancy is calculated in the same step as the scan-map, by selecting the passes in a binary decomposition-like order (e.g.: 0, 1, 1/2, 1/4, 3/4, 1/8, 3/8, 5/8, 7/8). The pass will be evaluated against the scan-map generated by all previously evaluated passes. A pass point declared redundant can be removed from the pass. While no pass will be entirely removed, the robotic manipulator can ”skip-ahead” to the next pass points at velocities unconstrained by NDI UT requirements.

The potential field process is run first, and when finished iterating, the redundant coverage step determines unused points. Once the redundant points are eliminated, the coverage path is complete. Example outcomes of the refinement process are shown in Fig. 6. The pass points form a raster-like pattern when the order of the pass points is alternated between passes. The pose and location of the pass points can be sent to any motion planning framework for collision avoidance and low-level motion planning. After verification by the operator, the path is sent to the robot for inspection.

V. Implementation and Results

The CPP algorithm was implemented on an NDI test cell with an ABB IRB-120 robotic manipulator and a Zivid 2+ structured light scanner. The algorithm was run in Python with the Open3d package [12] used to provide visual feedback to operators while the RoboDK SDK was used for motion planning and collision avoidance. Using a simple test piece, a comparison between the edge-fitting CPP algorithm and the edge propagation method from [2], seen in Fig. 1, was carried out with a 3.5 MHz, 64-element phased array ultrasonic probe.

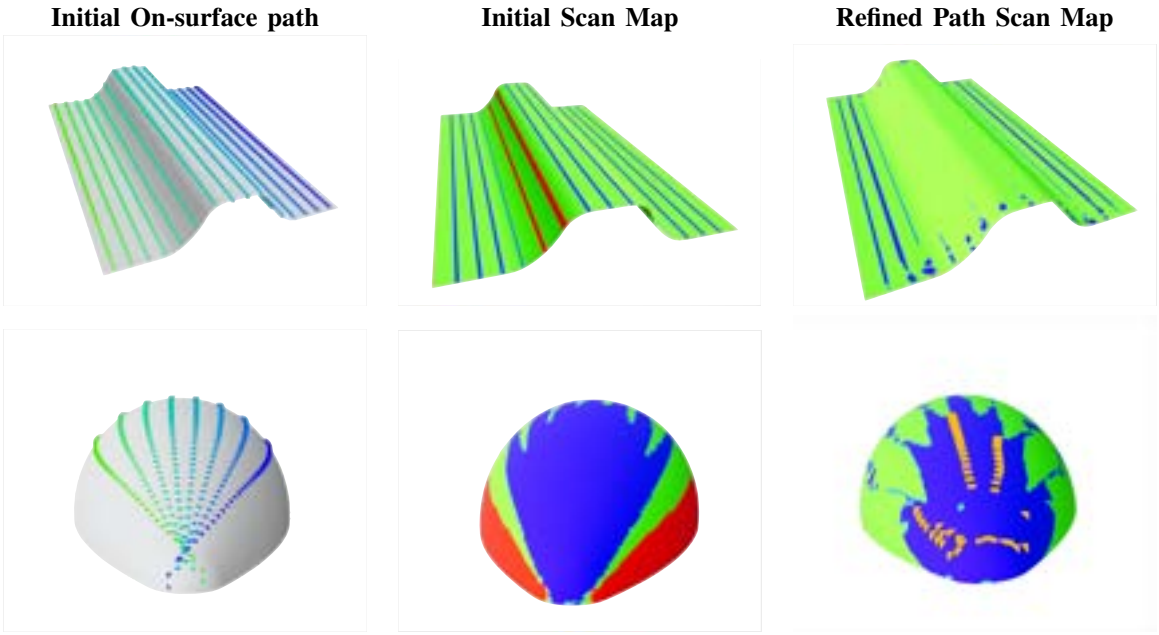


Fig. 6: Path refinement results demonstrated on a narrow stiffening structure and a degenerate nose-cone surface. The orange points on the Nose-cone are redundant and can be eliminated.

The results are shown in Fig.5, with the NDI data in purple overlaid on the optical scan of the test piece. While a slight mismatch between the scan and test-piece scan is expected due to the nature of NDI data, the red region in Fig. 5a highlights the data dropout experienced at the edge. Our proposed method performed as expected, achieving full coverage. Further simulation and testing is ongoing to provide metrics for coverage percentage, inspection time, and defect detection rate in order to quantitatively assess the method.

VI. Conclusion

In this paper, we introduced a novel coverage path planning method for robotic non-destructive inspection of composite parts using ultrasonic testing. Our approach addresses key limitations of existing methods by ensuring complete surface coverage, particularly at critical edges where traditional raster or sweep-based strategies fail. By incorporating Bézier curve fitting, we achieve accurate edge inspection, while a two-stage refinement algorithm works to optimize the coverage and minimize redundancy. The combination of these techniques results in a more efficient and reliable robotic inspection process. Future work may explore real-time path adjustments based on in-situ defect detection and extend the approach to other NDI techniques.

References

[1] P. Olivieri, L. Birglen, X. Maldague, and I. Mantegh, "Coverage path planning for eddy current inspection on complex aeronautical parts," *Robotics and Computer-Integrated Manufacturing*, vol. 30, no. 3, pp. 305–314, Jun. 2014.

[2] C. Mineo, S. G. Pierce, B. Wright, P. I. Nicholson, and I. Cooper, "Robotic path planning for non-destructive testing of complex shaped surfaces," 2015, pp. 1977–1987.

[3] M. Morozov, S. G. Pierce, C. N. MacLeod, C. Mineo, and R. Summan, "Off-line scan path planning for robotic NDT," *Measurement*, vol. 122, pp. 284–290, 2018.

[4] C. Mittelstedt, W. Becker, A. Kappel, and N. Kharghani, "Free-Edge Effects in Composite Laminates—A Review of Recent Developments 2005–2020," *Applied Mechanics Reviews*, vol. 74, no. 1, Jan. 2022.

[5] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258–1276, Dec. 2013.

[6] J. Song and S. Gupta, " :An Online Coverage Path Planning Algorithm," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 526–533, Apr. 2018.

[7] "Side-to-side 3d coverage path planning approach for agricultural robots to minimize skip/overlap areas between swaths," *Robotics and Autonomous Systems*, vol. 76, pp. 36–45, 2016.

[8] 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 on Automation Science and Engineering*, vol. 6, no. 1, pp. 107–120, Jan. 2009, conference Name: IEEE Transactions on Automation Science and Engineering.

[9] E. Galceran, R. Campos, N. Palomeras, D. Ribas, M. Carreras, and P. Ridao, "Coverage path planning with real-time replanning and surface reconstruction for inspection of three-dimensional underwater structures using autonomous underwater vehicles," *Journal of Field Robotics*, vol. 32, no. 7, pp. 952–983, 2015.

[10] K. Felsner, K. Schlachter, and S. Zambal, "Robotic Coverage Path Planning for Ultrasonic Inspection," *Applied Sciences*, vol. 11, no. 22, p. 10512, Nov. 2021.

[11] M. Shah, R. Eastman, and T. Hong, "An overview of robot-sensor calibration methods for evaluation of perception systems." 2012 Performance Metrics for Intelligent Systems Workshop, College Park, MD, US, 2012.

[12] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3d: A modern library for 3d data processing," *arXiv preprint arXiv:1801.09847*, 2018.

[13] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull, "Morse Decompositions for Coverage Tasks," *The International Journal of Robotics Research*, vol. 21, no. 4, pp. 331–344, Apr. 2002.

[14] J. Choi, R. Curry, and G. Elkaim, "Path Planning Based on Bézier Curve for Autonomous Ground Vehicles," in *Advances in Electrical and Electronics Engineering - IAENG Special Edition of the World Congress on Engineering and Computer Science 2008*. IEEE, pp. 158–166.