

Báo Cáo Nghiên Cứu Chuyên Sâu: Giải Pháp Thuật Toán Scan-to-Path Và Xử Lý Đám Mây Điểm Cho Hệ Thống Robot Hàn 6 Trục

1. Giới thiệu và Phạm vi Nghiên cứu

Trong bối cảnh tự động hóa công nghiệp hiện đại, việc tích hợp thị giác máy (machine vision) vào các hệ thống robot hàn 6 trục đã trở thành một yêu cầu cấp thiết nhằm nâng cao độ chính xác và khả năng thích ứng linh hoạt. Báo cáo này tập trung phân tích sâu vào quy trình công nghệ **Scan-to-Path** (Quét để tạo đường dẫn) sử dụng cảm biến Laser Profiler gắn trên tay máy (cấu hình Eye-in-Hand). Mục tiêu cốt lõi của nghiên cứu là giải quyết bài toán trích xuất đặc trưng hình học (Feature Extraction) cho hai loại mối hàn phổ biến nhất là mối hàn góc (Fillet) và mối hàn rãnh V (V-Groove), đồng thời xây dựng quy trình xử lý nhiễu dữ liệu đám mây (Point Cloud) và lập kế hoạch quỹ đạo 6 bậc tự do (6-DOF) bao gồm vị trí và hướng tiếp cận của mỏ hàn.

Báo cáo sẽ đi sâu vào các khía cạnh kỹ thuật phức tạp như xử lý nhiễu phản xạ quang học trên bề mặt kim loại, toán học của các phép biến đổi không gian để định hướng công cụ (Tool Center Point - TCP), và so sánh hiệu năng của các thư viện mã nguồn mở như Point Cloud Library (PCL) và Open3D trong môi trường công nghiệp. Các phân tích được tổng hợp từ dữ liệu nghiên cứu học thuật và thực tiễn triển khai, nhằm cung cấp một tài liệu tham chiếu toàn diện cho đội ngũ kỹ sư phát triển phần mềm điều khiển robot.

2. Kiến trúc Hệ thống Thu thập Dữ liệu và Cơ sở Lý thuyết

2.1. Cấu hình Eye-in-Hand và Nguyên lý Tam giác lượng Laser

Hệ thống Scan-to-Path dựa trên cấu hình Eye-in-Hand, trong đó cảm biến Laser Profiler (ví dụ: dòng Riftek RF627, Micro-Epsilon scanCONTROL) được gắn cố định vào khâu cuối (flange) của robot 6 trục.¹ Khác với cấu hình camera cố định, Eye-in-Hand cho phép mở rộng vùng làm việc không giới hạn và duy trì độ phân giải cao do khoảng cách từ cảm biến đến bề mặt phôi luôn được giữ ở mức tối ưu. Nguyên lý hoạt động dựa trên phương pháp tam giác lượng laser (Laser Triangulation), nơi một tia laser dạng vạch (line laser) chiếu lên bề mặt vật thể và hình ảnh biến dạng của vạch laser được thu nhận bởi cảm biến CMOS/CCD.

Dữ liệu thô thu được từ cảm biến là các lát cắt profile 2D (tọa độ x, z trong hệ quy chiếu cảm biến). Để tái tạo thành đám mây điểm 3D (Point Cloud), cần đồng bộ hóa dữ liệu này với vị trí của robot thông qua chuỗi biến đổi tọa độ động học thuận (Forward Kinematics).³

2.2. Chuỗi Biến đổi Tọa độ (Coordinate Transformation Chain)

Độ chính xác của hệ thống phụ thuộc hoàn toàn vào việc chuyển đổi chính xác tọa độ điểm đo từ không gian cảm biến sang không gian làm việc của robot (Base Frame). Một điểm P_{sensor} đo được bởi cảm biến sẽ được chuyển đổi sang hệ tọa độ gốc P_{base} theo công thức:

$$P_{base} = T_{base \rightarrow flange}(q) \cdot T_{flange \rightarrow sensor} \cdot P_{sensor}$$

Trong đó:

- $T_{base \rightarrow flange}(q)$: Ma trận biến đổi đồng nhất 4×4 từ đế robot đến mặt bích, phụ thuộc vào giá trị các khớp q tại thời điểm quét, được cung cấp bởi bộ điều khiển robot.⁵
- $T_{flange \rightarrow sensor}$: Ma trận biến đổi tĩnh từ mặt bích đến gốc cảm biến. Ma trận này phải được xác định thông qua quá trình hiệu chuẩn tay-mắt (Hand-Eye Calibration) với độ chính xác cao.⁴

Thách thức lớn nhất trong giai đoạn này không chỉ là độ chính xác cơ khí mà còn là sự đồng bộ về thời gian (temporal synchronization) giữa dữ liệu cảm biến và dữ liệu vị trí robot. Bất kỳ độ trễ nào cũng sẽ gây ra hiện tượng méo mó (distortion) trong đám mây điểm 3D tái tạo, đặc biệt khi robot di chuyển với vận tốc cao trong quá trình quét.

3. Quy trình Xử lý Nhiễu và Làm sạch Đám mây điểm

Dữ liệu đám mây điểm thu được từ quá trình quét bề mặt kim loại thường chứa nhiều nhiễu do đặc tính phản xạ gương (specular reflection) của vật liệu hàn, tia lửa (nếu quét online), và các yếu tố môi trường. Việc làm sạch dữ liệu là bước tiền xử lý bắt buộc để đảm bảo độ tin cậy cho các thuật toán trích xuất đặc trưng sau này.⁷

3.1. Phân loại Nhiễu trong Môi trường Hàn

Các loại nhiễu phổ biến bao gồm:

1. **Nhiễu lốm đốm (Speckle Noise):** Do sự giao thoa của ánh sáng laser kết hợp trên bề mặt nhám vi mô, tạo ra sai số ngẫu nhiên cục bộ.
2. **Nhiễu ngoại lai (Outliers):** Các điểm nằm xa bề mặt thực tế, thường do bụi, khói hàn,

- hoặc phản xạ biên cạnh.
3. **Nhiễu phản xạ gương (Reflective Noise):** Đây là loại nhiễu khó xử lý nhất trên kim loại bóng (nhôm, thép không gỉ). Tia laser bị phản xạ mạnh theo hướng khác thay vì khuếch tán về cảm biến, hoặc phản xạ đa đường (multipath) tạo ra các "bóng ma" dữ liệu nằm dưới bề mặt vật thể.⁷

3.2. Thuật toán Lọc Nhiễu Chuyên sâu

Để xử lý các loại nhiễu trên, sự kết hợp giữa các bộ lọc thống kê và bộ lọc dựa trên cường độ (intensity-based) là giải pháp tối ưu.

3.2.1. Statistical Outlier Removal (SOR)

Thuật toán loại bỏ ngoại lai thống kê (SOR) là tiêu chuẩn vàng để làm sạch nhiễu lốm đốm và các điểm bay lơ lửng. Cơ chế hoạt động dựa trên việc tính toán khoảng cách trung bình từ mỗi điểm đến k điểm lân cận gần nhất (k-Nearest Neighbors).⁹

Giả sử d_i là khoảng cách trung bình của điểm p_i đến k lân cận. Thuật toán tính giá trị trung bình μ_d và độ lệch chuẩn σ_d của toàn bộ tập hợp khoảng cách. Một điểm sẽ bị coi là nhiễu và loại bỏ nếu:

$$d_i > \mu_d + \alpha \cdot \sigma_d$$

Trong đó α là hệ số nhân độ lệch chuẩn (thường đặt từ 1.0 đến 2.0). Trong thư viện PCL, lớp `pcl::StatisticalOutlierRemoval` thực hiện chức năng này rất hiệu quả. Đối với Open3D, hàm `remove_statistical_outlier` cung cấp chức năng tương đương.¹²

3.2.2. Radius Outlier Removal (ROR)

ROR lọc các điểm dựa trên mật độ cục bộ. Người dùng định nghĩa một bán kính tìm kiếm R và số lượng điểm lân cận tối thiểu N_{min} . Nếu một điểm có ít hơn N_{min} lân cận trong bán kính R , nó được coi là điểm cô lập và bị loại bỏ. Phương pháp này đặc biệt hiệu quả để xóa các vật nhiễu rời rạc do khói hàn hoặc bụi.¹³

3.2.3. Lọc dựa trên Cường độ (Intensity Filter)

Đối với bề mặt kim loại có độ phản xạ cao, thông tin hình học đơn thuần là không đủ. Các cảm biến hiện đại trả về cả tọa độ (x, y, z) và cường độ phản xạ I . Các điểm nhiễu do phản xạ gương thường có cường độ bão hòa (quá cao) hoặc quá thấp (do tán xạ). Sử dụng bộ lọc điều kiện (`pcl::ConditionalRemoval` trong PCL) cho phép loại bỏ các điểm nằm ngoài dải cường độ

hợp lệ $[I_{min}, I_{max}]$, giúp loại bỏ đáng kể nhiễu phản xạ mà các bộ lọc không gian không phát hiện được.⁷

3.3. So sánh và Lựa chọn Thư viện: PCL vs Open3D

Việc lựa chọn công cụ phát triển ảnh hưởng trực tiếp đến hiệu năng và khả năng tích hợp của hệ thống. Dưới đây là bảng so sánh chi tiết giữa hai thư viện phổ biến nhất hiện nay.¹⁶

Tiêu chí	Point Cloud Library (PCL)	Open3D
Ngôn ngữ cốt lõi	C++ (Tối ưu hóa sâu)	C++ (Backend), Python (Frontend mạnh)
Kiến trúc	Modular, phức tạp, hướng đối tượng cổ điển.	Hiện đại, hỗ trợ Tensor, tích hợp tốt với DL.
Hiệu năng lọc	Rất cao nhờ tối ưu hóa OpenMP và cấu trúc dữ liệu KdTree.	Tốt, phiên bản mới hỗ trợ tăng tốc GPU (CUDA).
Trích xuất đặc trưng	Cực kỳ phong phú (NormalEstimation, PrincipalCurvatures, HarrisKeypoint3D).	Hạn chế hơn, tập trung vào hình học cơ bản và Deep Learning.
Tích hợp Công nghiệp	Tiêu chuẩn trong ROS, dễ đóng gói thành DLL C++/C# qua P/Invoke.	Phù hợp cho R&D nhanh, prototyping, AI pipelines.
Độ ổn định	Cao, đã được kiểm chứng qua hơn một thập kỷ.	Đang phát triển nhanh, API có thể thay đổi.

Kiến nghị: Đối với ứng dụng công nghiệp yêu cầu tính thời gian thực và ổn định trên hệ thống nhúng hoặc IPC, **PCL (C++)** là lựa chọn ưu tiên. Tuy nhiên, nếu hệ thống có tích hợp module Deep Learning để nhận diện loại mối hàn, Open3D sẽ là cầu nối tốt hơn.

4. Thuật toán Trích xuất Đặc trưng (Feature)

Extraction)

Đây là bước quan trọng nhất để chuyển đổi dữ liệu thô thành thông tin dẫn đường cho robot. Tùy thuộc vào loại mối hàn, chiến lược trích xuất sẽ khác nhau hoàn toàn.

4.1. Mối hàn Góc (Fillet Weld)

Mối hàn góc được tạo thành bởi hai tấm vật liệu ghép vuông góc (hoặc xiên góc). Về mặt hình học, đường hàn chính là giao tuyến của hai mặt phẳng.

4.1.1. Phân đoạn Mặt phẳng dùng RANSAC (Plane Segmentation)

Thuật toán RANSAC (Random Sample Consensus) là phương pháp mạnh mẽ nhất để giải quyết bài toán này do khả năng chống nhiễu ngoại lai xuất sắc.⁹

Quy trình thực hiện:

1. **Phát hiện Mặt phẳng 1:** Áp dụng RANSAC trên toàn bộ đám mây điểm (hoặc vùng ROI) để tìm mặt phẳng chiếm ưu thế nhất (thường là mặt đáy lớn). Mô hình toán học:
$$ax + by + cz + d = 0$$
2. **Tách inliers:** Loại bỏ tất cả các điểm thuộc mặt phẳng 1 (nằm trong ngưỡng distance_threshold) khỏi tập dữ liệu.
3. **Phát hiện Mặt phẳng 2:** Áp dụng RANSAC lần thứ hai trên tập điểm còn lại để tìm mặt phẳng thứ hai (mặt vách).
4. **Tính toán Giao tuyến:** Đường hàn lý thuyết là giao tuyến của hai mặt phẳng tìm được.

Vector chỉ phương của đường hàn u được tính bằng tích có hướng của hai pháp tuyến mặt phẳng: $u = n_1 \times n_2$. Điểm gốc P_{root} của đường hàn được tìm bằng cách giải hệ phương trình tuyến tính của hai mặt phẳng tại một tọa độ tham chiếu.

4.1.2. Tối ưu hóa RANSAC cho Mối hàn

Một biến thể "Improved RANSAC" được đề xuất trong các nghiên cứu gần đây⁹ sử dụng thông tin về độ cong bề mặt để định hướng quá trình lấy mẫu, thay vì lấy mẫu ngẫu nhiên thuần túy. Điều này giúp thuật toán hội tụ nhanh hơn và tránh việc khớp sai vào các bề mặt nhiễu. Ngoài ra, việc sử dụng ràng buộc về góc (ví dụ: hai mặt phẳng phải tạo với nhau một góc xấp xỉ 90 độ) giúp loại bỏ các kết quả dương tính giả.

4.2. Mối hàn Giáp mối Rãnh V (V-Groove Weld)

Khác với mối hàn Fillet, mối hàn V-Groove không thể được mô hình hóa đơn giản bằng hai mặt phẳng lớn do vùng rãnh hẹp và sự thay đổi hình học phức tạp tại đáy rãnh. Phương pháp tiếp cận hiệu quả nhất là phân tích lát cắt ngang (Cross-section/Scanline Analysis).²³

4.2.1. Phân tích Profile dựa trên Đạo hàm (Derivative-based Profile Analysis)

Vì dữ liệu từ Laser Profiler bản chất là tập hợp các đường quét (scanlines), ta có thể xử lý từng đường quét 2D độc lập trước khi tái hợp thành đường dẫn 3D.

1. **Làm trơn Profile:** Áp dụng bộ lọc Gaussian 1D hoặc Moving Average để giảm nhiễu tần số cao trên profile 2D.
2. **Tính Đạo hàm:** Tính đạo hàm bậc 1 (z') và bậc 2 (z'') của profile độ sâu theo trục ngang.
3. **Nhận diện Đặc trưng:**
 - o **Điểm vai (Shoulder points):** Được xác định tại vị trí đạo hàm bậc 1 thay đổi đột ngột vượt quá ngưỡng (bắt đầu dốc xuống).
 - o **Điểm đáy (Root point):** Là điểm cực trị địa phương nơi đạo hàm bậc 1 đổi dấu (từ dốc xuống sang dốc lên) và đạo hàm bậc 2 đạt giá trị cực đại (độ cong lớn nhất).³

4.2.2. Kỹ thuật Trích xuất 3D Skeleton (Skeletonization)

Đối với dữ liệu 3D toàn cục, có thể áp dụng phương pháp trích xuất khung xương (Skeletonization).²⁷ Phương pháp này sử dụng thuật toán co (contraction) để thu gọn đám mây điểm về trục trung tâm của nó. Đối với rãnh V, trục trung tâm này chính trùng với đường đáy mối hàn. Phương pháp này tính toán nặng hơn nhưng cho kết quả mượt mà hơn về tổng thể 3D.

5. Lập Kế hoạch Quỹ đạo và Tính toán Tư thế Robot (Path Planning & Robot Orientation)

Sau khi trích xuất được tập hợp các điểm đường hàn $P_{path} = \{p_1, p_2, \dots, p_n\}$, bước tiếp theo là xác định hướng của mỏ hàn tại từng điểm để tạo thành tập lệnh điều khiển 6 tham số (x, y, z, Rx, Ry, Rz) .

5.1. Hệ khung Frenet-Serret và Định hướng Mỏ hàn

Để đảm bảo chất lượng hàn, mỏ hàn phải duy trì một góc độ nhất định so với bề mặt vật liệu (Work Angle) và hướng di chuyển (Travel/Push/Drag Angle). Hệ khung Frenet-Serret di động gắn liền với đường cong hàn là công cụ toán học lý tưởng để giải quyết vấn đề này.²⁸

Tại mỗi điểm p_i trên đường hàn, ta xây dựng hệ trục trực chuẩn $\{T, N, B\}$:

1. **Vector Tiếp tuyến (T):** Xác định hướng di chuyển của robot.

$$T_i = \frac{p_{i+1} - p_{i-1}}{\|p_{i+1} - p_{i-1}\|}$$

2. **Vector Pháp tuyến Bề mặt (N_{surf})**: Đây là vector quan trọng để định hướng mỏ hàn.

Đối với mối hàn V-Groove, N_{surf} là trung bình cộng của pháp tuyến hai thành rãnh. Đối với Fillet, nó là vector phân giác của hai mặt phẳng tạo góc. Lưu ý rằng vector này khác với vector pháp tuyến chính (Principal Normal) của hệ Frenet thuần túy vốn hướng về tâm cong của đường đi.

3. **Vector Trùng pháp tuyến (B)**: Xác định bằng tích có hướng để hoàn thiện hệ trực phái:

$$B_i = T_i \times N_{surf}$$

Trục Z của mỏ hàn (Z_{tool}) – hướng dây hàn – thường được định nghĩa nằm trong mặt phẳng (N_{surf}, B) và tạo một góc nghiêng θ (Push/Drag angle) so với pháp tuyến bề mặt.

5.2. Chuyển đổi sang Biểu diễn Axis-Angle (Rx, Ry, Rz)

Hầu hết các robot công nghiệp như Universal Robots (UR) sử dụng biểu diễn **Rotation Vector** (hay Axis-Angle) để điều khiển hướng.⁵ Việc chuyển đổi từ ma trận quay mục tiêu sang Rotation Vector đòi hỏi sự chính xác toán học.

Quy trình chuyển đổi:

1. **Xây dựng Ma trận Quay Mục tiêu (R_{target})**: Từ các vector định hướng của mỏ hàn tính được ở bước trên (giả sử $X_{tool}, Y_{tool}, Z_{tool}$), ta lập ma trận quay 3x3:

$$R_{target} = \begin{bmatrix} | & | & | \\ X_{tool} & Y_{tool} & Z_{tool} \\ | & | & | \end{bmatrix}$$

2. **Chuyển đổi Rodrigues**: Sử dụng công thức Rodrigues hoặc các hàm thư viện (như cv::Rodrigues trong OpenCV hoặc phương thức GetRotationVector trong các thư viện robotics) để chuyển đổi R_{target} sang vector quay $r = \theta \cdot u$. Độ lớn $\theta = \|r\|$ là góc quay, và hướng $u = r / \theta$ là trục quay.³⁴

Lưu ý quan trọng về tính liên tục: Biểu diễn Axis-Angle có tính đa trị (ví dụ: góc quay α và $\alpha + 2\pi$ là tương đương về vật lý nhưng khác nhau về giá trị số). Khi tạo quỹ đạo liên tục, cần

kiểm tra và hiệu chỉnh để tránh hiện tượng robot xoay cổ tay đột ngột (wrist flip) 360 độ giữa hai điểm waypoint liền kề.

5.3. Làm trơn Quỹ đạo (Trajectory Smoothing)

Dữ liệu quét thực tế luôn có nhiều tần số cao khiến đường dẫn bị rung. Trước khi gửi xuống bộ điều khiển robot, quỹ đạo cần được làm trơn:

- **Vị trí:** Sử dụng nội suy B-Spline hoặc bộ lọc Savitzky-Golay để làm mượt tọa độ (x, y, z) .³⁶
- **Hướng:** Sử dụng nội suy cầu (Slerp - Spherical Linear Interpolation) trên Quaternion trước khi chuyển đổi ngược về Axis-Angle. Điều này đảm bảo sự thay đổi hướng mỏ hàn diễn ra mượt mà, đồng đều dọc theo đường hàn.

6. Tổng hợp Mã nguồn Tham khảo và Hướng dẫn Triển khai

Để hỗ trợ quá trình phát triển, việc tham khảo các mã nguồn mở đã được kiểm chứng là rất cần thiết. Dưới đây là các tài nguyên và cấu trúc lớp (class) cụ thể trong PCL và Open3D nên được sử dụng.

6.1. Các Repo GitHub Tiêu biểu

1. [indraneelpatil/3D-Weld-Seam-Tracking-using-PCL](#):
 - **Mô tả:** Đây là một dự án mẫu mực sử dụng PCL C++ để xử lý đám mây điểm cho robot 5-6 trục. Nó bao gồm các module về lọc nhiễu, phân đoạn mặt phẳng và tính toán đường tâm mối hàn.³⁷
 - **Điểm nhấn:** Logic trích xuất đặc trưng hình học được triển khai rõ ràng trong file `weld_seam.cpp`.
2. [thillRobot/seam_detection](#):
 - **Mô tả:** Một gói phần mềm (package) tích hợp sẵn với ROS (Robot Operating System), cung cấp các node C++ để phát hiện đường hàn. Dự án này bao gồm cả quy trình đăng ký (registration) và chuyển đổi tọa độ TF.³⁹
 - **Ứng dụng:** Phù hợp nếu hệ thống của bạn sử dụng ROS làm middleware kết nối robot và cảm biến.
3. [PatilVrush/3D_Weld_Seam_Detection_and_Tracking_using_PCL_ROS_ANN](#):
 - **Mô tả:** Dự án này mở rộng việc phát hiện đường hàn bằng cách kết hợp PCL với Mạng nơ-ron nhân tạo (ANN). Tuy nhiên, phần tiền xử lý PCL vẫn là giá trị cốt lõi để tham khảo cho việc làm sạch dữ liệu.⁴⁰

6.2. Các Lớp C++ Quan trọng trong PCL

Khi triển khai bằng C++, các lớp sau đây của PCL là không thể thiếu:

- `pcl::StatisticalOutlierRemoval<pcl::PointXYZ>`: Dùng cho bước lọc nhiễu đầu tiên.¹⁰
- `pcl::SACSegmentation<pcl::PointXYZ>`: Dùng để thực hiện RANSAC tìm mặt phẳng. Cần thiết lập `setModelType(pcl::SACMODEL_PLANE)` và `setMethodType(pcl::SAC_RANSAC)`.¹⁹
- `pcl::PrincipalCurvaturesEstimation<pcl::PointXYZ, pcl::Normal, pcl::PrincipalCurvatures>`: Dùng để tính toán độ cong tại mỗi điểm, hỗ trợ trích xuất biên dạng rãnh V hoặc tinh chỉnh kết quả RANSAC.⁴¹
- `pcl::NormalEstimation<pcl::PointXYZ, pcl::Normal>`: Bước tiên quyết để tính toán hướng bề mặt cho feature extraction.⁴²

6.3. Kiến trúc Phần mềm Đề xuất

Một kiến trúc phần mềm mạnh mẽ cho tính năng Scan-to-Path nên được chia thành các module độc lập:

1. **Sensor Interface**: Driver giao tiếp với Laser Profiler (thường qua TCP/IP hoặc UDP), chịu trách nhiệm nhận gói tin profile và giải mã.
2. **Data Reconstruction**: Module nhận dữ liệu encoder robot và ghép nối các profile 2D thành Point Cloud 3D.
3. **Processing Pipeline (PCL Core)**:
 - Input: Raw Point Cloud.
 - Step 1: ROI Filtering (Passthrough filter).
 - Step 2: Noise Removal (SOR + Intensity filter).
 - Step 3: Feature Extraction (RANSAC/Slice Analysis).
 - Output: List of Waypoints (x, y, z, i, j, k) .
4. **Path Planner**: Chuyển đổi Waypoints + Normal Vectors thành Robot Pose (x, y, z, Rx, Ry, Rz) và nội suy quỹ đạo.
5. **Robot Comm**: Gửi quỹ đạo xuống bộ điều khiển robot (qua Profinet, Ethernet/IP hoặc script socket).

7. Kết luận và Khuyến nghị

Phát triển tính năng Scan-to-Path cho robot hàn là một bài toán đa ngành, đòi hỏi sự kết hợp chặt chẽ giữa thị giác máy tính và điều khiển robot. Kết quả nghiên cứu chỉ ra rằng:

1. **Về Xử lý Nhiễu**: Không thể chỉ dựa vào một loại bộ lọc. Cần phối hợp `StatisticalOutlierRemoval` để loại bỏ nhiễu ngẫu nhiên và `ConditionalRemoval` dựa trên cường độ (Intensity) để xử lý triệt để nhiễu phản xạ trên bề mặt kim loại bóng.
2. **Về Trích xuất Đặc trưng**:
 - Đối với **Fillet**, RANSAC tìm giao tuyến mặt phẳng là giải pháp bền vững và chính xác nhất.

- Đối với **V-Groove**, phương pháp phân tích đạo hàm trên profile 2D (Scanline derivative) kết hợp làm trơn B-Spline mang lại hiệu quả cao hơn so với xử lý 3D thuần túy.
3. **Về Công cụ: PCL (C++)** được khuyến nghị mạnh mẽ cho môi trường sản xuất công nghiệp do hiệu năng và sự ổn định. Open3D thích hợp hơn cho giai đoạn nghiên cứu thuật toán và kiểm thử nhanh (prototyping).
 4. **Về Tạo Quỹ đạo:** Việc áp dụng hệ khung **Frenet-Serret** kết hợp với công thức **Rodrigues** để chuyển đổi sang Axis-Angle là chìa khóa để tạo ra các đường hàn mượt mà, đúng kỹ thuật cho robot 6 trục.

Việc tuân thủ quy trình này sẽ giúp hệ thống đạt được độ chính xác dưới milimet (sub-millimeter accuracy), đáp ứng các tiêu chuẩn khắt khe của ngành hàn tự động.

Tài liệu tham khảo tích hợp: ¹

Works cited

1. Laser Seam Tracking System for Welding Automation, accessed February 1, 2026, https://fae.it/cms/Laser%20Seam%20Tracking%20System%20for%20Welding%20Automation?language_code=ENG
2. LASER SEAM TRACKING SYSTEM FOR WELDING AUTOMATION RF627Weld Series, accessed February 1, 2026, https://riftek.com/upload/iblock/e95/Laser_Seam_Tracking_System_for_Welding_Automation_eng.pdf
3. Laser Scanning and Parametrization of Weld Grooves with Reflective Surfaces - MDPI, accessed February 1, 2026, <https://www.mdpi.com/1424-8220/21/14/4791>
4. Sensor-guided robotic laser welding - https://ris.utwente.nl/ws/files/6086329/thesis_de_Graaf.pdf
5. How Can I Acquire Rx, Ry Rz? - Technical Questions - Universal Robots Forum, accessed February 1, 2026, <https://forum.universal-robots.com/t/how-can-i-acquire-rx-ry-rz/37987>
6. Robot Tool Center Point Calibration using Computer Vision - Diva-portal.org, accessed February 1, 2026, <https://www.diva-portal.org/smash/get/diva2:23964/FULLTEXT01.pdf>
7. Reflective Noise Filtering of Large-Scale Point Cloud Using Multi-Position LiDAR Sensing Data - MDPI, accessed February 1, 2026, <https://www.mdpi.com/2072-4292/13/16/3058>
8. Laser Scanning and Parametrization of Weld Grooves with Reflective Surfaces, accessed February 1, 2026, https://www.researchgate.net/publication/353222135_Laser_Scanning_and_Parametrization_of_Weld_Grooves_with_Reflective_Surfaces
9. Weld seam recognition algorithm based on a fast point cloud plane fitting method, accessed February 1, 2026, <https://opg.optica.org/josaa/abstract.cfm?uri=josaa-43-2-307>

10. pcl::StatisticalOutlierRemoval< PointT > Class Template Reference - Point Cloud Library, accessed February 1, 2026,
https://pointclouds.org/documentation/classpcl_1_1_statistical_outlier_removal.html
11. Removing outliers using a StatisticalOutlierRemoval filter - Point Cloud Library, accessed February 1, 2026,
https://pointclouds.org/documentation/tutorials/statistical_outlier.html
12. PointCloud - Open3D 0.19.0 documentation, accessed February 1, 2026,
https://www.open3d.org/docs/release/tutorial/t_geometry/pointcloud.html
13. StatisticalOutlierRemoval | pcl.js, accessed February 1, 2026,
<https://pcl.js.org/docs/api/filters/statistical-outlier-removal>
14. pcl::ConditionalRemoval< PointT > Class Template Reference - Point Cloud Library, accessed February 1, 2026,
https://pointclouds.org/documentation/classpcl_1_1_conditional_removal.html
15. open3d.t.geometry.PointCloud, accessed February 1, 2026,
https://www.open3d.org/docs/latest/python_api/open3d.t.geometry.PointCloud.html
16. Point Cloud Libraries: PCL vs. Open3D for Processing Efficiency - Patsnap Eureka, accessed February 1, 2026,
<https://eureka.patsnap.com/article/point-cloud-libraries-pcl-vs-open3d-for-processing-efficiency>
17. A curated list of awesome Point Cloud Processing Resources, Libraries, Software - GitHub, accessed February 1, 2026,
<https://github.com/mmolero/awesome-point-cloud-processing>
18. Performance comparisons against PCL and Open3D in common operations.... - ResearchGate, accessed February 1, 2026,
https://www.researchgate.net/figure/Performance-comparisons-against-PCL-and-Open3D-in-common-operations-Left-column-running_fig2_328374976
19. Welding Line Detection Using Point Clouds from Optimal Shooting Position, accessed February 1, 2026,
<https://omu.repo.nii.ac.jp/record/2001442/files/2024000783.pdf>
20. An Identification and Localization Method for 3D Workpiece Welds Based on the DBSCAN Point Cloud Clustering Algorithm - MDPI, accessed February 1, 2026,
<https://www.mdpi.com/2504-4494/8/6/287>
21. A weld feature extraction method based on lightweight instance segmentation and point cloud features | Industrial Robot | Emerald Publishing, accessed February 1, 2026,
<https://www.emerald.com/ir/article/doi/10.1108/IR-07-2025-0236/1329990/A-weld-feature-extraction-method-based-on>
22. (PDF) A feature-extraction localization algorithm research for teaching-free automated robotic welding based on 3D point cloud - ResearchGate, accessed February 1, 2026,
https://www.researchgate.net/publication/385264245_A_feature-extraction_localization_algorithm_research_for_teaching-free_automated_robotic_welding_based_on_3D_point_cloud

23. A Novel Seam Tracking Technique with a Four-Step Method and Experimental Investigation of Robotic Welding Oriented to Complex Welding Seam - MDPI, accessed February 1, 2026, <https://www.mdpi.com/1424-8220/21/9/3067>
24. Laser Scanning and Parametrization of Weld Grooves with Reflective Surfaces - PMC, accessed February 1, 2026, <https://pmc.ncbi.nlm.nih.gov/articles/PMC8309933/>
25. Automatic Groove Measurement and Evaluation with High Resolution Laser Profiling Data - Semantic Scholar, accessed February 1, 2026, <https://pdfs.semanticscholar.org/0387/da58669536d904964c1a7d7d32993b6364d7.pdf>
26. Development of a characteristic point detecting seam tracking algorithm for portable welding robots - ResearchGate, accessed February 1, 2026, https://www.researchgate.net/publication/224207820_Development_of_a_characteristic_point_detecting_seam_tracking_algorithm_for_portable_welding_robots
27. Two-Dimensional Skeleton Intersection Extraction-Based Method for Detecting Welded Joints on the Three-Dimensional Point Cloud of Sieve Nets - MDPI, accessed February 1, 2026, <https://www.mdpi.com/2073-8994/17/9/1484>
28. Understanding the Frenet-Serret frame | by Sakshi Kakde - Medium, accessed February 1, 2026, <https://sakshik.medium.com/understanding-the-frenet-serret-frame-3b9c730e8b1c>
29. Trajectory Planning in the Frenet Space - Robotics Knowledgebase, accessed February 1, 2026, <https://roboticsknowledgebase.com/wiki/planning/frenet-frame-planning/>
30. 1 The Differential Geometry of Curves - Robotics, accessed February 1, 2026, <http://robotics.caltech.edu/~jwb/courses/ME115/handouts/curves>
31. Difference between RX RY RZ angle between robot and what sent by profinet, accessed February 1, 2026, <https://forum.universal-robots.com/t/difference-between-rx-ry-rz-angle-between-robot-and-what-sent-by-profinet/30001>
32. Axis-angle representation - Universal Robots, accessed February 1, 2026, <https://www.universal-robots.com/articles/ur/programming/axis-angle-representation/>
33. rotvec2rpy(rotation_vector) - Universal Robots, accessed February 1, 2026, https://www.universal-robots.com/manuals/EN/HTML/SW10_7/Content/prod-scriptmanual/G5/rotvec2rpy_rotation_vector.htm
34. Rotation matrix - Wikipedia, accessed February 1, 2026, https://en.wikipedia.org/wiki/Rotation_matrix
35. Standard matrix of a rotation on a vector in \mathbb{R}^3 - Math Stack Exchange, accessed February 1, 2026, <https://math.stackexchange.com/questions/3040112/standard-matrix-of-a-rotation-on-a-vector-in-mathbbR^3>
36. Efficient and Accurate Start Point Guiding and Seam Tracking Method for Curve Weld Based on Structure Light - ResearchGate, accessed February 1, 2026, https://www.researchgate.net/publication/350779434_Efficient_and_Accurate_St

- [art_Point_Guiding_and_Seam_Tracking_Method_for_Curve_Weld_Based_on_Structure_Light](#)
37. 3D-Weld-Seam-Tracking-using-PCL/weld_seam_tracking.cp at master - GitHub, accessed February 1, 2026,
https://github.com/indraneelpatil/3D-Weld-Seam-Tracking-using-PCL/blob/master/weld_seam_tracking.cp
38. [indraneelpatil/3D-Weld-Seam-Tracking-using-PCL](#): In this project we have used the Point Cloud Library - GitHub, accessed February 1, 2026,
<https://github.com/indraneelpatil/3D-Weld-Seam-Tracking-using-PCL>
39. [thillRobot/seam_detection](#): ROS package for weld seam detection using pointcloud data - GitHub, accessed February 1, 2026,
https://github.com/thillRobot/seam_detection
40. [PatilVrush/3D_Weld_Seam_Detection_and_Tracking_using_PCL_ROS_ANN](#): In this project an algorithm was developed to autonomously detect 3D weld seams on workpiece of any geometrical shape using Point Cloud Library. Further we used Neural Networks to trace the detected weld seam using a 5 DOF Robotic Manipulator. Other projects like autonomous_grasping using feature matching are also included. - GitHub, accessed February 1, 2026,
https://github.com/PatilVrush/3D_Weld_Seam_Detection_and_Tracking_using_PCL_ROS_ANN
41. [pcl::PrincipalCurvaturesEstimation< PointInT, PointNT, PointOutT > Class Template Reference](#) - Point Cloud Library, accessed February 1, 2026,
http://pointclouds.org/documentation/classpcl_1_1_principal_curvatures_estimation.html
42. [pcl/examples/features/example_principal_curvatures_estimation.cpp](#) at master - GitHub, accessed February 1, 2026,
https://github.com/otherlab/pcl/blob/master/examples/features/example_principal_curvatures_estimation.cpp
43. X-Y-Z-Rx-Ry-Rz-position. – UR Forum-Help-Q&A, accessed February 1, 2026,
<https://www.zacobria.com/universal-robots-knowledge-base-tech-support-forum-hints-tips-cb2-cb3/index.php/x-y-z-rx-ry-rz-position/>
44. Filtering Pointclouds for SLAM - Aditya Kamath, accessed February 1, 2026,
<https://adityakamath.github.io/2021-04-19-pointcloud-laserscan-filters/>
45. [jbruening/PclSharp](#): Point Cloud Library pinvoke binding for c - GitHub, accessed February 1, 2026, <https://github.com/jbruening/PclSharp>
46. A new fast filtering algorithm for a 3D point cloud based on RGB-D information | PLOS One, accessed February 1, 2026,
<https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0220253>
47. Explanation on robot orientation - Universal Robots, accessed February 1, 2026,
<https://www.universal-robots.com/articles/ur/application-installation/explanation-on-robot-orientation/>
48. Point cloud — Open3D 0.13.0 documentation, accessed February 1, 2026,
<https://www.open3d.org/docs/0.13.0/tutorial/geometry/pointcloud.html>