

Path Planning With Automatic Seam Extraction Over Point Cloud Models for Robotic Arc Welding

Peng Zhou^{ID}, *Student Member, IEEE*, Rui Peng, Maggie Xu^{ID}, Victor Wu,
and David Navarro-Alarcon^{ID}, *Senior Member, IEEE*

Abstract—This letter presents a point cloud based robotic system for arc welding. Using hand gesture controls, the system scans **partial** point cloud views of workpiece and **reconstructs** them into a **complete 3D model** by a linear **iterative closest point** algorithm. Then, a **bilateral filter** is extended to denoise the workpiece model and preserve important geometrical information. To extract the welding seam from the model, a novel **intensity-based** algorithm is proposed that **detects edge points** and generates a **smooth** 6-DOF welding path. The methods are tested on multiple workpieces with different joint types and poses. Experimental results prove the robustness and efficiency of this robotic system on automatic path planning for welding applications.

Index Terms—Industrial robots, motion and path planning, RGB-D perception.

I. INTRODUCTION

INDUSTRIAL welding robots have been extensively studied with the aim to solve labor shortage problem and improve productivity. Currently, “teach and playback” mode or offline programming prevails **in robotic welding workshops**. As these conventional practices lack adaptation to accommodate changes in a working environment, many advanced techniques such as seam detection and tracking are gradually being employed in industry.

Solutions to intelligent robotic arc welding primarily require the assistance of computer vision. Previous studies have conducted in-depth research on image processing and established viable methods across several visual sensors, including mono camera, stereo camera, and CCD camera [1]–[6]. [7] developed an automatic image acquisition system for real-time weld

pool analysis and seam tracking. The seam characteristics were measured by defining region of interest and implementing pattern learning algorithm. [8] proposed a shape-matching algorithm for an online laser-based seam detection. The algorithm enabled autonomous detection for different groove types and localizes the boundary of weld seam. [9] measured the cross-sectional areas of 2D profiles and finds joint features and volume of the workpiece.

While above mentioned studies can effectively locate the seam, their implementations are only validated on simple or flat welding workpieces with common joint types. In most cases, the groove problem is regarded as two-dimensional and image processing is sufficient to identify the target. However, it is important to discuss the generalization of these techniques to cope with complex geometries. The remaining challenge lies in capturing **free-form grooves**. Point cloud processing is favored as it can comprehensively describe complex 3D shapes. A captured point cloud model can reveal useful geometric information but simultaneously introduce considerable **noise**. The **complexity and computational cost** also evolve as the information increases. Therefore, designing a **robust** algorithm that can efficiently utilise 3D information will be crucial for this application. Although, some recent studies using point cloud processing were able to detect the groove, they did not further exploit the features to improve system functionality, such as plan a welding path with **well-defined orientation** [10]–[14]. More experiments on complex and diverse seam patterns are needed for verification. In addition, the authors found that most current systems and commercial products need to manually set or program the initial poses for visual sensors (laser sensor, etc.) to scan the workpiece. This guidance of scanning may be realized in a more intuitive way to facilitate human machine collaboration.

To address the above problems, we propose a novel point cloud based system for robotic welding planning. This system integrates a single RGB-D camera with a cooperative industrial manipulator to fulfill the generic requirements for seam detection. Guided by hand gestures, the system captures a set of partial point clouds of the workpiece from different perspectives. To reconstruct the surface, it adopts a linear iterative closest point algorithm that improves the rate of fitness and extends **bilateral filter** algorithm for 3D surface denoising. A novel algorithm is implemented to **identify the groove based on edge density**. After the localization, the system plans a **smooth consecutive welding path** with 6-DOF and executes it with the robot manipulator. To

Manuscript received November 23, 2020; accepted March 25, 2021. Date of publication April 2, 2021; date of current version April 20, 2021. This letter was recommended for publication by Associate Editor Y. Joo and Y. Choi upon evaluation of the reviewers' comments. This work was supported in part by the Research Grants Council under Grant 14203917, in part by the Chinese National Engineering Research Centre for Steel Construction Hong Kong Branch under Grant BBV8, in part by the Key-Area Research and Development Program of Guangdong Province 2020 (project 76), and in part by PolyU under Grants YBYT and ZZHJ. (*Corresponding author: David Navarro-Alarcon.*)

Peng Zhou, Maggie Xu, Victor Wu, and David Navarro-Alarcon are with the Department of Mechanical Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong (e-mail: jeffery.zhou@connect.polyu.hk; xinrxu@polyu.edu.hk; victor.wh.wu@gmail.com; dna@ieee.org).

Rui Peng is with the The Hong Kong Polytechnic University, Kowloon, Hong Kong, and also with the University of Hong Kong, Pok Fu Lam, Hong Kong (e-mail: u3008069@connect.hku.hk).

This letter has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2021.3070828>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2021.3070828

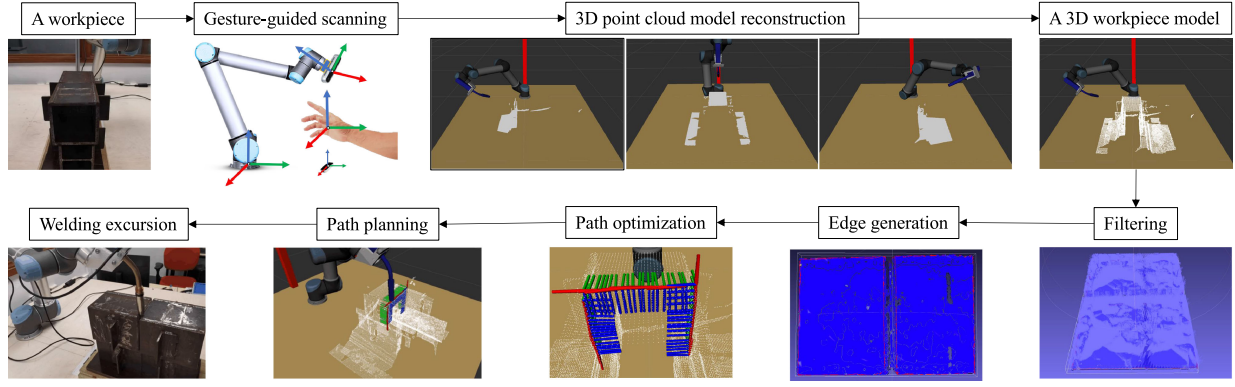


Fig. 1. Overview of the automatic robotic welding pipeline. Given a workpiece, a gesture-guided scanning is executed to merge captured point clouds into a completed 3D model, which is further improved with the bilateral filter. Subsequently, on basis of the edge points detected using an edge intensity gradient-based algorithm, a 6-DOF welding path is established.

validate the proposed methods, we conduct detailed experiments on workpieces with different joint types.

The remaining parts of this article are as follows. Section II introduces the architecture of the proposed planning system and the key techniques applied. The experimental results of the proposed system are presented in Section III. Section IV gives final conclusions.

II. METHODOLOGY

A. System Overview

The robotic system architecture is depicted in Fig. 1. The complete **surface** model reconstruction for the target welding workpiece, which a hand gesture-based method, is designed to interact with the manipulator with assistance from a human operator. Subsequently, the depth camera, which is mounted on the end-effector of the robot, moves to several poses to capture different views of the point cloud. Then, the point cloud model of the workpiece is reconstructed by combining all the individual frames of the point cloud for the workpiece. On the basis of the complete surface model, **the gradient of the edge intensity-based welding seam detection algorithm is employed to locate the targeted weld path points**. With these generated path points, an interpolation of both translation and rotation can be used to generate a 6-DOF welding path, which can then be sent to the robot manipulator. Finally, the robot executes the path tracking motion to finish the welding tasks on the targeted workpiece.

B. Hand Gesture Guided Scanning

To scan the welding workpiece, hand gestures are captured for generating control commands to move the camera. As shown in Fig. 2, we used a real-world setup to best represent the commonly used cases in this research. A **“joystick” interaction mode** is designed to interact with a robot, on which a depth camera is mounted. This interaction process is handled by the virtual interaction box above the hand gesture sensor. The moving direction of the scanning depth camera is determined according to the right hand’s relative position with respect to the interaction box, and the rotation of the camera is mapped with the raw,

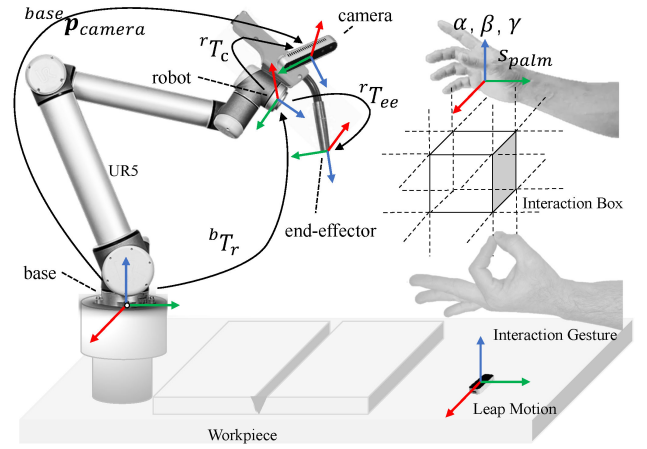


Fig. 2. Conceptual representation of the intuitively interactive scanning with hand gestures, in which an interaction box for right hand is built to define different mapping rules for robotic motions and “ok” gesture for left hand to capture the point cloud data.

pitch and yaw of right-hand gesture captured with leap motion. Meanwhile, the gesture of the left hand is used to control the point cloud capture and the pose of the robot end-effector at the current poses for the succeeding reconstruction process.

We present the entire interaction process in Alg. 1. In the frame of the UR manipulator, a standard pose is formulated with $\mathbf{p} = (\mathbf{s}, \mathbf{r})$ where \mathbf{s} is 3D position represented by (x, y, z) and \mathbf{r} is a rotation vector denoted by (r_x, r_y, r_z) . The algorithm takes the position of the right hand’s palm position s_{palm} , the hand’s Euler angles (α, β, γ) , and the current pose with respect to the camera $base \mathbf{p}_{camera}^t$ as the inputs. The resulting desired output is target pose $base \mathbf{p}_{camera}^{t+1}$ in the next time step. The entire interaction is in a loop of an infinite hand frame set $\{\mathcal{F}_{hand}^i, i = t\}$ with respect to time step t captured by leap motion sensor. In each frame \mathcal{F}_{hand}^i , the right hand’s palm position s_{palm} and its hand’s Euler angles (α, β, γ) are mapped into motions of the robotic manipulator based on the region in which the right hand stays with respect to the interaction box above the sensor. According to the relative position with respect to the interaction box, the entire interaction space is separated into seven regions, namely, home,

Algorithm 1: Right Hand Gesture Interaction.

```

1 if True then
2    $(s_{palm}^t, \alpha, \beta, \gamma) = \text{getPalmAttribute}(\mathcal{F}_{hand}^t)$ 
3   if  $s_{palm}^t$  not in home region then
4      $s_{camera}^{t+1} += \mathbf{u}_d \cdot l$ 
5      $\mathbf{p}_{camera}^{t+1} = (s_{camera}^{t+1}, \mathbf{r}_{camera}^t)$ 
6   end
7   else
8      ${}^{lm}\mathbf{R}_{hand} = \text{Transform}(\alpha, \beta, \gamma)$ 
9      ${}^{base}\mathbf{R}_{hand} = {}^{base}\mathbf{R}_{lm} \cdot {}^{lm}\mathbf{R}_{hand}$ 
10     $\mathbf{r}_{camera}^{t+1} = \text{InvRodrigues}({}^{base}\mathbf{R}_{hand})$ 
11     $\mathbf{p}_{camera}^{t+1} = (s_{camera}^t, \mathbf{r}_{camera}^{t+1})$ 
12  end
13  return  $\mathbf{p}_{camera}^{t+1}$ 
14 end

```

left, right, up, down, inward, and outward regions. When s_{palm} reaches other regions except the home region, a corresponding directional unit vector \mathbf{u}_d and distance weight l (remain the displacement in an appropriate range) are computed to update the position part of the target pose ${}^{base}\mathbf{p}_{camera}^{t+1}$. However, if s_{palm} stays in the home region, the manipulator switches into rotation mode, then the Eule angles captured by the gesture sensor are to be transformed into rotation matrix ${}^{lm}\mathbf{R}_{hand}$, then this rotation matrix will be left multiplied by a rotation matrix from the gesture sensor to the robot base coordinate system ${}^{base}\mathbf{R}_{lm}$. With the resulting rotation matrix ${}^{base}\mathbf{R}_{hand}$, the inverse *Rodrigues* formula is used to transform it into the corresponding rotation part $\mathbf{r}_{camera}^{t+1}$ to update ${}^{base}\mathbf{p}_{camera}^{t+1}$. During the process, the left hand's gesture is used to store the point cloud data \mathcal{P}_i and the corresponding camera pose with respect to the base when a certain "ok" gesture is detected.

C. Welding Workpiece Reconstruction

In performing a high precision reconstruction for the welding workpieces, an initial reconstruction is executed on the basis of the scanned point cloud set, $\{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n\}$, and their calculated camera pose, $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$, and then a linear iterative closest point (ICP) algorithm [15] is employed to further improve the fitness performance of the workpiece model. For each point cloud \mathcal{P} , the transformation from camera coordinate frame, ${}^c\mathbf{F}_{\mathcal{P}}$, to the base frame, ${}^b\mathbf{F}_{\mathcal{P}}$, is given by:

$${}^b\mathbf{F}_{\mathcal{P}} = {}^b\mathbf{T}_r \cdot {}^r\mathbf{T}_c \cdot {}^c\mathbf{F}_{\mathcal{P}} \quad (1)$$

where ${}^r\mathbf{T}_c$ is a fixed rigid transformation from the depth camera to robot obtained by hand-eye calibration [16], and ${}^b\mathbf{T}_r$ can be easily computed using the inverse kinematics module. To simplify, the origin of the global coordinate system is set at the base of the robot. By performing the transformations illustrated in Eq. (1) on the dataset $\{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n\}$, the entire point cloud data has been set in the base coordinate system, then they can be combined as a completed point cloud $\{{}^b\mathcal{P}_1 \cup {}^b\mathcal{P}_2 \cup \dots \cup {}^b\mathcal{P}_n\}$ for the welding workpiece. In further improving the fitness, an ICP process setting the transformation from the depth camera to robot base frame, ${}^b\mathbf{T}_c$, as its initial configuration is

needed. Specifically, the i -th source point is defined as $\mathbf{s}_i = (s_{ix}, s_{iy}, s_{iz}, 1)^T$, its corresponding target point is represented by $\mathbf{t}_i = (t_{ix}, t_{iy}, t_{iz}, 1)^T$, and the unit normal vector at \mathbf{t}_i can be denoted as $\mathbf{n}_i = (n_{ix}, n_{iy}, n_{iz}, 0)^T$. The goal of each ICP iteration is to obtain \mathbf{T}_* , that is:

$$\mathbf{T}_* = \arg \min_{\mathbf{T}} \sum_i ((\mathbf{T}_o \cdot \mathbf{s}_i - \mathbf{t}_i) \cdot \mathbf{n}_i)^2 \quad (2)$$

where \mathbf{T}_* and \mathbf{T}_o are 4×4 3D target and initial transformation matrices. In its first iteration, ${}^b\mathbf{T}_c$ is set as an initial transform matrix to accelerate the convergence. Usually, a rotation matrix $\mathbf{R}(\alpha, \beta, \gamma)$ and a translation matrix $\text{Trans}(p_x, p_y, p_z)$ can be decomposed into a 3D transformation \mathbf{T} , which can be denoted as:

$$\mathbf{T} = \text{Trans}(p_x, p_y, p_z) \cdot \mathbf{R}(\alpha, \beta, \gamma) \quad (3)$$

Eq. 2 is a essentially a least-square optimization problem whose solution normally requires the determination of the six parameters of $\alpha, \beta, \gamma, p_x, p_y$, and p_z . However, in $\mathbf{R}(\alpha, \beta, \gamma)$, three of them (α, β and γ) correspond to the arguments of non-linear trigonometric functions. Consequently, we cannot employ an efficient linear least-square technique to solve this problem. Nonetheless, as the relative orientation between the source surface and the target surface decreases after each ICP iteration, the linear approximation will become more accurate in the next running, and $\alpha, \beta, \gamma \approx 0$ will be satisfied finally. Therefore, the approximations $\sin \theta \approx \theta$ and $\cos \theta \approx 1$ can be achieved, thus changing the original nonlinear least-square problem into a linear one. In this manner, a linear least-square algorithm can still be carried out as follows:

$$\hat{\mathbf{R}}(\alpha, \beta, \gamma) \approx \begin{pmatrix} 1 & -\gamma & \beta & 0 \\ \gamma & 1 & -\alpha & 0 \\ -\beta & \alpha & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4)$$

$$\hat{\mathbf{T}} = \text{Trans}(p_x, p_y, p_z) \cdot \hat{\mathbf{R}}(\alpha, \beta, \gamma) \quad (5)$$

$$\hat{\mathbf{T}}_* = \arg \min_{\hat{\mathbf{T}}} \sum_i ((\hat{\mathbf{T}} \cdot \mathbf{s}_i - \mathbf{t}_i) \cdot \mathbf{n}_i)^2 \quad (6)$$

where $(\hat{\mathbf{T}} \cdot \mathbf{s}_i - \mathbf{t}_i)$ can be written as a linear expression with six parameters ($\alpha, \beta, \gamma, p_x, p_y$, and p_z). Then, N pairs of point correspondences can be arranged as a linear matrix expression as follows:

$$\min_{\hat{\mathbf{T}}} \sum_i ((\hat{\mathbf{T}} \cdot \mathbf{s}_i - \mathbf{t}_i) \cdot \mathbf{n}_i)^2 = \min_{\mathbf{x}} |\mathbf{A}\mathbf{x} - \mathbf{b}|^2 \quad (7)$$

Thus, $\hat{\mathbf{T}}_*$ can be obtained by solving:

$$\mathbf{x}_* = \arg \min_{\mathbf{x}} |\mathbf{A}\mathbf{x} - \mathbf{b}|^2 \quad (8)$$

which can be further solved using singular value decomposition (SVD).

D. Welding Seam Extraction

To remove noise from the point cloud data without removing vital edge information, an extension of bilateral filter [17]

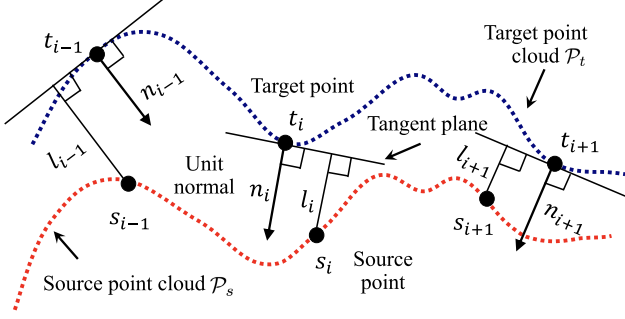


Fig. 3. Conceptual representation of the ICP process from a source point cloud \mathcal{P}_s to a target point cloud \mathcal{P}_t .

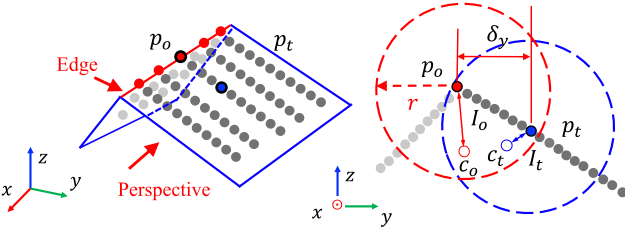


Fig. 4. Conceptual representation of the edge intensity and gradient along the Y-axis for current point p_o , in which c_o represents the local geometric centroid for p_o and edge intensity I_o with respect to p_o is higher than I_t with respect to its neighbor p_t . Gradient I_{oy} is defined as the difference between I_o and I_t divided by the coordinate difference along the Y-axis.

specific for 3D point cloud data is used. As the point density of a reconstructed workpiece model varies, few points of a neighborhood exist within a fixed range. The establishment of a minimum point m will provide a rational neighborhood to reduce the effect on points density. Therefore, for a given point $p \in \mathcal{P}$, its neighboring point q can be defined as:

$$\mathcal{N}_r(p) = \{q \in \mathcal{P} \mid \|q - p\|_2 < r \cup |\mathcal{N}| \geq m\} \quad (9)$$

where r is the fixed radius, which can adaptively adjust to guarantee that no less than m points will exist in the neighborhood of p . As shown in Fig. 4, the normalized displacement between original point p_o and its local geometric centroid c_o can be defined as the edge intensity of p_o , which is calculated by:

$$I_o = \frac{1}{r} \left\| p_o - \frac{1}{n} \sum_{i=0}^n q_i \right\| \quad (10)$$

where $\frac{1}{n} \sum_{i=0}^n q_i$ denotes the geometric centroid for all the neighboring points corresponding to original point p_o . Within the edge-free areas, the intensity remains low, but the value will increase when approaching edges and reaches a local maximum in the intensity space at edges. Fig. 4 shows a side view of two intersecting point cloud surfaces as an example, in which the edge intensity I_o of edge point p_o is much higher compared with the intensity I_t of its neighbor p_t in non-edge region, as the distance between the local geometric centroid c_o and p_o is larger than the distance between p_t and c_t .

For 2D images, the gradients typically represent the direction in which the features of a pixel mostly change. The way of defining gradients for 3D point clouds can represent the

variations between edge intensities within its neighborhood. However, unlike the gradient decomposition in consecutive 2D pixels, gradients for a 3D point cloud can only be approximated due to the discrete points. On the basis of the 3D gradient operators [18], we define the 3D gradient decompositions for a point p_o as:

$$I_k \approx \max_{i \in \mathcal{N}} \left[(I_i - I_o) \cdot \frac{d_{i,o}}{\delta k} \right], k = x, y, z \quad (11)$$

where δx , δy , and δz are the coordinate differences in three axes, and $d_{i,o}$ represents the distance between a current point p_o and its i -th neighboring point q_i . Fig. 4 shows an example of the point's gradient along Y axis.

A 3D offset function $F(x, y, z)$ for point clouds analogous with the edge response function [19] for 2D pixels is presented and approximated by a Taylor series expansion. For a given point p_o , the offset function $F(x_o, y_o, z_o)$ can be expressed as a multiplication of a small offset $(\delta x, \delta y, \delta z)$ and a symmetric matrix \mathbf{H} , in which I_x , I_y , and I_z can be calculated by:

$$\begin{aligned} F(x_o, y_o, z_o) &= \sum_{\delta x, \delta y, \delta z} [I_{x_o + \delta x, y_o + \delta y, z_o + \delta z} - I_{x, y, z}]^2 \\ &= \sum_{\delta x, \delta y, \delta z} \left[\delta x \cdot I_x + \delta y \cdot I_y + \delta z \cdot I_z + O(\delta x^2, \delta y^2, \delta z^2) \right]^2 \\ &\approx (\delta x, \delta y, \delta z) \mathbf{H} (\delta x, \delta y, \delta z)^\top \end{aligned} \quad (12)$$

$$\mathbf{H} = \begin{bmatrix} I_x^2 & I_x I_y & I_x I_z \\ I_y I_x & I_y^2 & I_y I_z \\ I_z I_x & I_z I_y & I_z^2 \end{bmatrix} \quad (13)$$

where the distribution of the gradients for the local neighborhood is depicted by the eigenvalues of \mathbf{H} . Therefore, the intensity of gradient changes can be detected by comparing the difference of eigenvalues. For example, a high eigenvalue reveals a large shift in gradient along the related eigenvector, and this rule is widely used to classify the extracted features for 2D pixels. In extracting edges in point clouds, two observations will be detected, namely, 1) the co-existence of two high eigenvalues, which can indicate an edge lying on an intersection between two point cloud surfaces, and 2) a single existence of a high eigenvalue, which can reveal a boundary of the point cloud. In the context of welding applications, both of the situations will be considered because the welding seam only lies on the intersection between two surfaces or the boundary between different workpieces. Additionally, as ‘‘high’’ eigenvalues with a threshold are hard to describe, the ratios between them are measured to identify a threshold t instead of solely using the magnitude of eigenvalues. Most importantly, this ratio calculation scheme does not rely on a time-consuming SVD.

Inspired by approach in [19] to avoid the calculation of eigenvalues, we use $\lambda_1 = \alpha \lambda_3$ and $\lambda_2 = \beta \lambda_3$ ($\alpha \geq \beta \geq 1$) to represent the three sorted eigenvalues ($\lambda_1 \geq \lambda_2 \geq \lambda_3$). Subsequently, the ratio between trace $\text{Tr}(\mathbf{H})$ and determinant $\text{Det}(\mathbf{H})$ can be calculated using 14. As the ratio increases with larger α and β , we can set a threshold t to guarantee the existence of

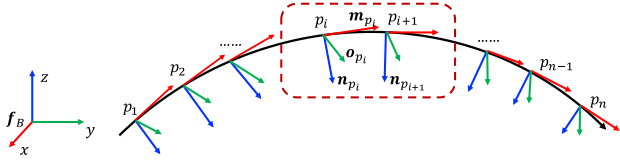


Fig. 5. Determination of three unit orthogonal basis vectors of the 3-DOF rotation for each welding seam pose.

large eigenvalues:

$$\frac{\text{Tr}(\mathbf{H})^3}{\text{Det}(\mathbf{H})} = \frac{(\lambda_1 + \lambda_2 + \lambda_3)^3}{\lambda_1 \lambda_2 \lambda_3} = \frac{(\alpha + \beta + 1)^3}{\alpha \beta} \geq t \quad (14)$$

$$G(o, i) = \frac{1}{\delta \sqrt{2\pi}} e^{-\frac{\|p_o - q_i\|^2}{2\delta^2}}, \quad q_i \in \mathcal{N} \quad (15)$$

Then, Eq. (15) is used to smoothen \mathbf{H} and ensure $\text{Det}(\mathbf{H})$ is a non-zero, in which σ is the standard deviation of the distance between current p_o and its neighborhood $\mathcal{N}_r(p_o)$. Finally, a set of edge points $\{p_1^{(e)}, p_2^{(e)}, \dots, p_n^{(e)}\}$ is generated.

E. Welding Path Optimization

On the basis of the extracted edge points, $p_i^{(w)} = p_i^{(e)} + d * \mathbf{n}_{p_i^{(e)}}$ is used to refine the edge point to welding path point for guiding the welding torch, where d is the optimal distance to weld a workpiece suggested by different sets of welding equipment. Once the 3-DOF translation parameters $\{p_1^{(w)}, p_2^{(w)}, \dots, p_n^{(w)}\}$ for welding path points are determined, 3-DOF rotation parameters are necessary to form a complete welding path. The computation of rotation parameters needs to define three unit orthogonal basis vectors with respect to the robot base frame. As shown in Fig. 5, the first vector is defined as the pointing vector of the welding torch, which equals to the normal vector of the surrounding welding region point cloud. This approach aligns with the basic requirement of keeping the welding torch perpendicular to the welding region. The second vector is the moving direction of the welding torch, which is defined as the vector \mathbf{v}_m between two nearest path points. Note that rotating freely around the welding direction is not considered because it will not improve the welding quality, but, by contrast, it will add the path planning complexity for welding seams. As for vertical welding cases, either vertical-up or vertical-down direction is acceptable because the robot manipulator can maintain an appropriate speed to prevent burn-through. Furthermore, to reduce the disturbance of the moving direction, its final moving vector is defined as the average vector of the m nearest moving vectors, i.e., $\mathbf{m}_i = (\sum_{i-\frac{m}{2}}^{i+\frac{m}{2}} \mathbf{v}_{m_i})/m$. According to the right-hand rule, the third vector is computed via cross-multiplication, i.e., $\mathbf{o}_i = \mathbf{m}_i \times \mathbf{n}_i$. Therefore, each path point has its own coordinate that is formed by $(\mathbf{o}_i, \mathbf{m}_i, \mathbf{n}_i)^T$.

In view of achieving superior welding motion performance, increasing the density of path points is necessary to support the robotic manipulator in smoothly executing the path-tracking motion [20]. A linear path interpolation method is designed from two aspects, namely, 3D translation interpolation and 3D rotation interpolation. In particular, a number of ρ points is

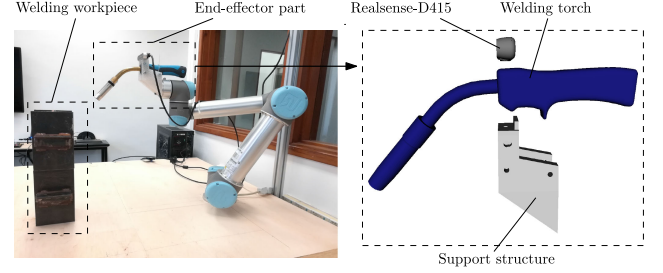


Fig. 6. (Left) Experimental welding platform. (Right) End-effector of the proposed welding robotic system comprising an RGB-D camera and welding torch. The camera is positioned above the welding torch.

interpolated between $p_i^{(w)}$ and $p_{i+1}^{(w)}$ and the j -th interpolation point is defined as:

$$p_j^{(w_i)} = p_i^{(w)} + j \cdot \frac{p_{i+1}^{(w)} - p_i^{(w)}}{\rho} \quad (16)$$

Then, each initial path point $p_i^{(w)}$ owns its rotation vector $\mathbf{v}_i^{(w)}$ and the j -th rotation vector is interpolated between $\mathbf{v}_i^{(w)}$ and $\mathbf{v}_{i+1}^{(w)}$, which is computed with the spherical-linear rotation interpolation method:

$$\mathbf{v}_j^{(w_i)} = \frac{\sin((1-\alpha)\theta)\mathbf{v}_i^{(w)} + \sin(\alpha\theta)\mathbf{v}_{i+1}^{(w)}}{\sin\theta} \quad (17)$$

where $\alpha = j/\rho$ and $\theta = \arccos(\mathbf{v}_i^{(w)}, \mathbf{v}_{i+1}^{(w)})$. Finally, a dense and even 6-DOF path for welding is planned.

III. RESULTS

In evaluating the performance of the proposed welding path planning method, we built a robotic system based on UR5 and tested it on several workpieces. To verify the robustness, two complicated types (cylinder and Y-shaped workpieces) as shown in Fig. 9 are selected in each following step of the entire planning process for illustration. Both of the two workpieces are widely applied in offshore structures and long-span space trusses, which are challenging tasks because it requires welders to work aloft and combat gravity. Fig. 6 has previously shown the experimental platform and the RGB-D sensor adopted in this work. The result of conventional teach-playback robotic path planning method, which is regarded the benchmark of welding path planning, is also obtained to compare with our proposed approach.

A. Gesture-Guide Scanning

The gesture-guide scanning is implemented via leap motion SDK v3.2 and Python SDK for UR robots, namely, URX. To avoid the effect of shaking hands, we averaged the hand's movement, orientation and gesture in the past 10 frames. The interaction box is designed as a spatial cube with a length of 10 cm edge, which is appropriate for the interactions in this study. To validate the performance of scanning the workpiece with bare hands, the experiment uses right hand palm position to guide the depth sensor compete the movement from one

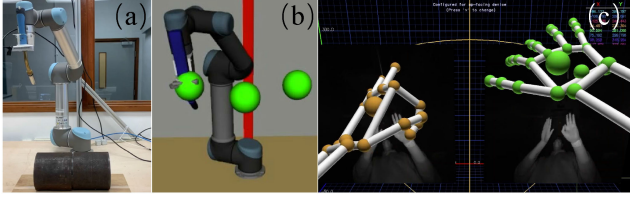


Fig. 7. (a) Real environment. (b) Three waypoints in Rviz. (c) Hand gestures in Leap Motion visualizer.

TABLE I
PERFORMANCE OF LEAP MOTION SCANNING

Point/Path	Action	Success Rate	Cost Time(s)	Error(cm)
Home A	Move	100%	3.45	-
A	Capture	100%	1.62	3.2
AB	Move	100%	6.13	-
B	Capture	100%	1.21	2.9
BC	Move	90%	5.37	-
C	Capture	100%	1.56	4.3

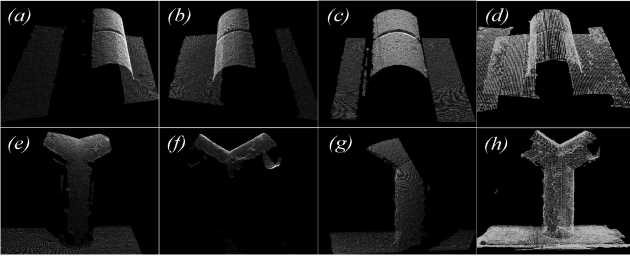


Fig. 8. Model reconstruction. (a)–(c), (e)–(g) Partial point clouds from different views. (d, h) Merge partial point clouds to form a completed model.

point to another target point, and left hand gestures to execute the command of capturing point clouds. As shown in Fig 7, we set three waypoints A, B, and C for the depth sensor to capture the point cloud from left to right. With gestures, the depth sensor is moved from the home position to reach A, B, C. During the process, the time and errors in the operation is recorded. In this experiment, 10 operators who had never operated a robot were selected to conduct the experiment after 10 minutes's training. The cost time of moving between different points and the distance between capture points and target points were obtained respectively. For capture points, it is considered successful when the distances to the target points are within 5 cm and the captured point clouds cover the whole workpiece. While for moving paths, it is considered successful when the robot moves between different points without any collisions. From the Table I we can observe that the maximum average errors at three points are 4.3 cm, the moving paths Home-A and AB both success, only once operation happens a collision when moving along BC path. Generally, the robustness is good.

B. Workpiece Reconstruction

As shown in Fig. 8, partial point clouds from different perspectives are generated during the scanning process. The camera pose with respect to the world frame is calculated using a forward kinematics module provided by URX and a fixed transformation from the camera to the robot end-effector. On

the basis of this information, a preliminary 3D model of the workpieces are reconstructed, as shown in Figs. 8(d) and (h). In measuring the reconstruction performance, a metric of the fitness score is computed as follows:

$$S(\mathbf{M}) = \frac{|\mathcal{P}_m|}{|\mathcal{P}_t|}, \left\{ m_i \in \mathcal{P}_m \mid \|t_i - \mathbf{M}m_i\|^2 < d \right\} \quad (18)$$

where t_i and m_i are the target point and the model point, respectively, and \mathbf{M} is the iteratively optimized transformation. Therefore, this score can measure the ratio of successfully aligned points. As depicted by the trend in Fig. 10, the fitness score of the cylinder piece represented by the red spline is higher than the other Y-shaped piece labeled with the black spline throughout the entire ICP process. Before the ICP, the model of the cylinder piece was 86%, and that of the Y-shaped piece was 81%, but these values are not acceptable in high-precision welding considering that 94% and above can ensure the average distance between corresponding points is within 0.2 mm. During the ICP process, the cylinder piece quickly reached the convergence when the iteration equalled 5, while the Y-shaped piece needed 15 iterations to reach the convergence. This trend can be explained by the Y-shaped piece that has a more complex geometrical surface than the cylinder piece.

C. Welding Path Detection

After rasterization, the harmonic mean of accuracy is taken to measure the overall performance. The mean is defined as:

$$F_1 = \frac{3}{a_{xy}^{-1} + a_{xz}^{-1} + a_{yz}^{-1}} \quad (19)$$

where a_{xy}^{-1} , a_{xz}^{-1} and a_{yz}^{-1} denote the accuracy calculated for the xy , xz , and yz planes after rasterization. The corresponding rasterized point clouds are shown in Figs. 12(e) and (f). Each detected edge point is displayed in red, and the ground truth path point is manually marked in green based on point cloud generated by a high-precision 3D scanner. The overall performance calculated on the basis of the coverage of the red edge point over the ground truth path rasterized on different planes. The effect of the bilateral filter on the welding path detection also needs to be determined. Figs. 11(a) and (b) show the overall pre- and post-filtering performances of the two selected workpieces, respectively. The figures show that the performances of the two workpieces both have increased after filtering. This trend is due to the filtering, which reduces the noise and smoothens the surface for the point cloud data. Fig. 10 validates this difference based on the normal map of the pre- and post-filtering processes of the workpieces, especially on the corresponding regions surrounded by dashed circles. Besides, different predefined thresholds t have an impact on the overall performance. Figs. 12(a)–(c) show the extracted edges when t is equal to 10, 70, and 50, respectively. If the t is too small or too large, then it will negatively affect the overall performance score. The cylinder has reached the peak when its threshold is near 53, whereas the Y-shaped piece needs a slightly larger threshold of approximately 60 to attain the peak performance.

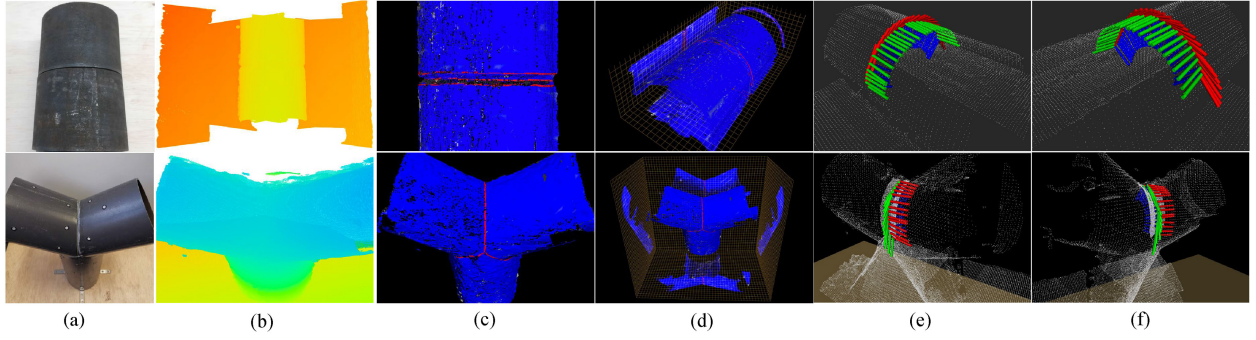


Fig. 9. Results of different processes in the proposed system. (a) Two actual welding metal workpieces, cylinder and Y-shape. (b) Reconstructed models of workpieces with a depth map. (c) Welding seam extracted with the edge intensity gradient-based algorithm. (d) Rasterized views of extracted edge points. (e, f) 6-DOF welding path after interpolation.

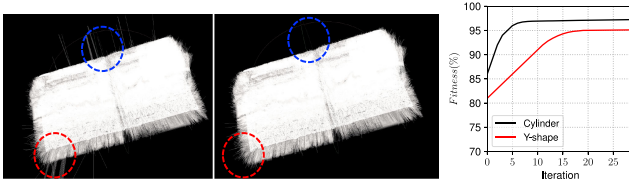


Fig. 10. (Left) Comparative results of pre and post-denoising with the bilateral filter. (Right) Fitness score of two workpieces with different iterations.

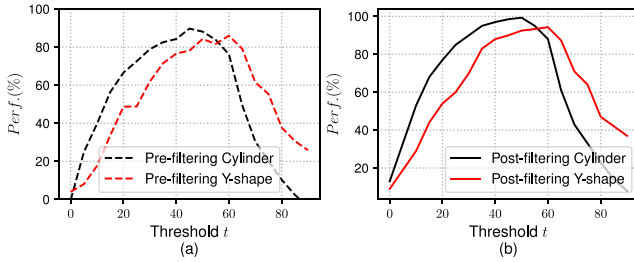


Fig. 11. Pre- and post-performance scores under different thresholds.

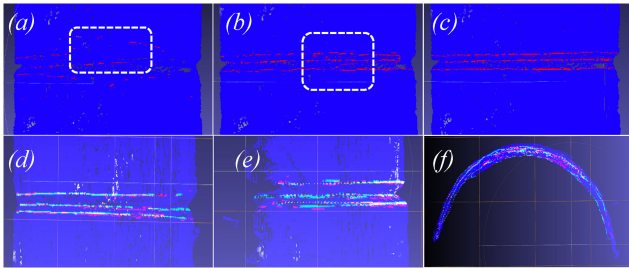


Fig. 12. (a)–(c) Overall performance with $t = 10, 70, 50$, (d)–(f) Rasterization to measure the precision of the generated welding path.

D. Pose Optimization

To measure the accuracy of the optimized welding path generated by our system, a series of evenly distributed welding waypoints (both 30 points for all the workpieces) was set by an experienced welder in the teach-playback method. Normally, each waypoint requires 5 seconds to 10 seconds to set. Then, a random sample consensus (RANSAC) algorithm was applied to estimate a smooth curve with a mathematical model from these waypoints. Circle and ellipse equations are selected to estimate the cylinder and Y-shaped workpieces, respectively. To keep the

TABLE II
ACCURACY FOR EACH WORKPIECE

System	Types	Error(mm)	X	Y	Z
Ours	Cylinder	ME	1.24	1.39	1.56
Ours	Cylinder	RMSE	0.61	0.67	0.77
Ours	Y-shaped	ME	1.47	1.63	1.74
Ours	Y-shaped	RMSE	0.74	0.80	0.85
Ours	Flat (straight)	ME	1.28	1.45	1.52
Ours	Flat (straight)	RMSE	0.59	0.63	0.67
Ref [22]	Flat (straight)	ME	1.36	1.69	1.31
Ref [22]	Flat (straight)	RMSE	0.64	0.67	0.73s

calculation simple, the ground truth paths are all calibrated in the zy plane. Given a welding point $m_i(x_i, y_i, z_i)$ generated by our model, the corresponding ground truth point $g_i(0, p_i, q_i)$ is defined as the nearest point whose distance to the tangent line is the smallest (meanwhile, this tangent line is perpendicular to the line between these two points). Considering the cylinder workpiece as an example, $g_i(0, p_i, q_i)$ can be solved by

$$\begin{aligned} (p_i - a)^2 + (q_i - b)^2 &= r^2, q_i > b \\ -\frac{(p_i - a)(z_i - q_i)}{(q_i - b)(y_i - p_i)} &= -1 \end{aligned} \quad (20)$$

where a, b and r are given by the RANSAC model. Then, the root mean square error (RMSE) along with x, y , and z -axis between the two path points is used to evaluate the accuracy of the proposed method. The RMSE is defined as:

$$RMSE(m, g) = \sqrt{\frac{1}{n} \sum_{i=1}^n (m_i - g_i)^2} \quad (21)$$

where $\{m_i\}$ represents the generated welding path by our model, $\{g_i\}$ is the ground truth of the welding path, and n denotes the number of path points.

According to the principle of arc welding for constructions [21], an error within 1mm will not affect the welding quality. To compare with the state-of-the-artwork, the result of a straight v-type flat workpiece is added. The RMSE results of the workpieces are shown in Table II, which reveals acceptable accuracy for the welding path generated by our system.

TABLE III
RUNTIME ON V-TYPE BUTT JOINT WELDING PATH PLANNING

System	Joint Pose	Method	Runtime (ms)	Speed (mm/s)
Ours	Flat	Point cloud	3691	135
Ours	Tilted	Point cloud	3891	129
Ours	Horizontal	Point cloud	3600	139
Ours	Vertical	Point cloud	4100	122
Classical	Flat	Teach-playback	12600	\
Ref [9]	Flat	Laser vision	\	5
Ref [20]	Flat	Point cloud	14090	\
Ref [22]	Flat	Point cloud	4913	\

E. Comparison With Existing Methods

Compared with previous studies on groove detection, **thresholding eigenvalue of gradient** makes system performance less vulnerable to noise in complex environments and thus our system can be flexibly applied to a wide range of workpiece. This is supported by sufficient experiments from plane placed at various poses to complex 3d shapes. However, the output welding path is computed based on previous downsampled point cloud which may fail to maintain concrete local geometric information, especially for points at the corners. Accuracy is another weakness of the system as the standard value given by other studies [1]–[3], [7] is under 0.5 mm. Instead of using manual controls or pre-programming poses for positioning the camera, we adopt a gestured based interaction for scanning. Combined with a broad view vision sensor, the system can reconstruct free form workpieces and plan the path at a competitively fast speed. We measured the runtime for the whole system on a 500 mm straight V-type butt joint as this type is commonly presented in the literature. The full workpiece can be reconstructed from one capture. The average time for 20 runs is taken for comparison and the detection speed defined by length of seam over runtime is calculated. The results listed in Table III show that our system outputs consistent results across different settings and is significantly efficient for seam detection.

IV. CONCLUSION

This article presents a sensor-guided welding robotic system [23]. Specifically, an interactive algorithm based on gesture is first proposed to scan the workpiece in an intuitive manner. Then, the system reconstructs a 3D point cloud model of the target by a linear ICP algorithm. The precision and speed of the reconstruction is ensured by taking the pose obtained from robotic kinematics as the initial setting in ICP. Next, after filtering, the welding seam joints is robustly identified using **a gradient-based edge detection approach**. Based on the detected seam, **a smooth 6-DOF path is computed for execution**. Experiments on various geometric shapes have validated the effectiveness of the proposed system. In the future, physical welding experiments will be conducted with our proposed system. Additionally, we may introduce other new interaction techniques to assist robotic welding tasks.

REFERENCES

- [1] Y. Xu *et al.*, “Real-time seam tracking control technology during welding robot gta process based on passive vision sensor,” *J Mate Process Tech.*, vol. 212, no. 8, pp. 1654–1662, 2012.
- [2] J. Fan *et al.*, “An initial point alignment and seam-tracking system for narrow weld,” *IEEE Trans. Industr. Inform.*, vol. 16, no. 2, pp. 877–886, Feb. 2020.
- [3] T. Font comas, C. Diao, J. Ding, S. Williams, and Y. Zhao, “A passive imaging system for geometry measurement for the plasma arc welding process,” *IEEE Trans. Ind. Electron.*, vol. 64, no. 9, pp. 7201–7209, Sep. 2017.
- [4] X. Li, X. Li, S. S. Ge, M. O. Khyam, and C. Luo, “Automatic welding seam tracking and identification,” *IEEE Trans. Ind. Electron.*, vol. 64, no. 9, pp. 7261–7271, Sep. 2017.
- [5] H. N. M. Shah *et al.*, “Review letter on vision based identification, detection and tracking of weld seams path in welding robot environment,” *Mod. App. Sci.*, vol. 10, no. 2, pp. 83–89, 2016.
- [6] A. Rout, B. Deepak, and B. Biswal, “Advances in weld seam tracking techniques for robotic welding: A review,” *Robot. Cim-Int. Manuf.*, vol. 56, pp. 12–37, 2019.
- [7] L. Nele, E. Sarno, and A. Keshari, “An image acquisition system for real-time seam tracking,” *Int. J. Adv. Manuf. Tech.*, vol. 69, no. 9-12, pp. 2099–2110, 2013.
- [8] Y. Ding, W. Huang, and R. Kovacevic, “An on-line shape-matching weld seam tracking system,” *Robot. Cim-Int. Manuf.*, vol. 42, pp. 103–112, 2016.
- [9] R. Manorathna *et al.*, “Feature extraction and tracking of a weld joint for adaptive robotic welding,” in *Proc. Int. Conf. Cont. Auto. Robot Vis.*, 2014, pp. 1368–1372.
- [10] L. Zhang, Y. Xu, S. Du, W. Zhao, Z. Hou, and S. Chen, “Point cloud based three-dimensional reconstruction and identification of initial welding position,” in *Trans. Intel. Weld. Manuf.*, 2018, pp. 61–77.
- [11] S. M. Ahmed, Y. Z. Tan, C. M. Chew, A. A. Mamun, and F. S. Wong, “Edge and corner detection for unorganized 3 d point clouds with application to robotic welding,” in *Proc. IEEE/RSJ Int. Conf. Robots Intell. Syst.*, 2018, pp. 7350–7355.
- [12] V. Patil *et al.*, “Extraction of weld seam in 3d point clouds for real time welding using 5 dof robotic arm,” in *Proc. Int. Conf. Cont., Auto. Robot*, 2019, pp. 727–733.
- [13] L. Jing, J. Fengshui, and L. En, “Rgb-d sensor-based auto path generation method for arc welding robot,” in *Proc. Chin. Control Decis. Conf.*, 2016, pp. 4390–4395.
- [14] K. Zhang, M. Yan, T. Huang, J. Zheng, and Z. Li, “3D reconstruction of complex spatial weld seam for autonomous welding by laser structured light scanning,” *J. Manuf. Process*, vol. 39, pp. 200–207, 2019.
- [15] K.-L. Low, “Linear least-squares optimization for point-to-plane icp surface registration,” *Chapel Hill, Univ. North Carolina*, vol. 4, no. 10, pp. 1–3, 2004.
- [16] R. Y. Tsai and R. K. Lenz, “A new technique for fully autonomous and efficient 3D robotics hand/eye calibration,” *IEEE Trans. Robot. Autom.*, vol. 5, no. 3, pp. 345–358, Jun. 1989.
- [17] J. Digne and C. De Franchis, “The bilateral filter for point clouds,” *Image Process. On Line*, vol. 7, pp. 278–287, 2017.
- [18] Y. Zhang, “Quantitative study of 3d gradient operators,” *Image Vis. Comput.*, vol. 11, no. 10, pp. 611–622, 1993.
- [19] C. G. Harris *et al.*, “A combined corner and edge detector,” in *Proc. Alvey Vis. Conf.*, vol. 15, no. 50, 1988, pp. 10–5244.
- [20] R. Peng, D. Navarro-Alarcon, V. Wu, and W. Yang, “A point cloud based method for automatic groove detection and trajectory planning of robotic arc welding tasks,” in *Proc. IEEE Int. Conf. Ubiquitous Robots*, 2020, pp. 380–386.
- [21] L. E. Company, *The Procedure Handbook of Arc Welding*, 2000.
- [22] L. Yang *et al.*, “A novel system for off-line 3D seam extraction and path planning based on point cloud segmentation for arc welding robot,” *Robot. Cim-Int. Manuf.*, vol. 64, 2020, Art. no. 101929.
- [23] A. Cherubini and D. Navarro-Alarcon, “Sensor-based control for human-robot collaboration: Fundamentals, challenges and opportunities,” *Front Neurobotics*, vol. 14, 2021, Art. no. 113.