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

Peng Zhou, Student Member, IEEE, Rui Peng, Maggie Xu, Victor Wu, and David Navarro-Alarcon, 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 localized 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.
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, 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 $p = (s, r)$ where $s$ is 3D position represented by $(x, y, z)$ and $r$ 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 $(\alpha, \beta, \gamma)$, and the current pose with respect to the camera base $p_{base}$ as the inputs. The resulting desired output is target pose $p_{camera}^{t+1}$ in the next time step. The entire interaction is in a loop of an infinite hand frame set $\{F_{hand, i} = t\}$ with respect to time step $t$ captured by leap motion sensor. In each frame $F_{hand}$, the right hand’s palm position $s_{palm}$ and its hand’s Euler angles $(\alpha, \beta, \gamma)$ 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   \[ (s_{\text{palm}}, \alpha, \beta, \gamma) = \text{getPalmAttribute}\left(F_{\text{hand}}^i\right) \]
3 if \( s_{\text{palm}} \) not in home region then
4     \[ s_{\text{camera}}^{i+1} = +u_d \cdot l \]
5     \[ p_{\text{camera}}^{i+1} = (s_{\text{camera}}^{i+1}, r_{\text{camera}}^{i+1}) \]
6 end
7 else
8     \[ \text{lm\_R}_{\text{hand}} = \text{Transform}(\alpha, \beta, \gamma) \]
9     \[ \text{base\_R}_{\text{hand}} = \text{base\_R}_{\text{lm}} \cdot \text{lm\_R}_{\text{hand}} \]
10    \[ r_{\text{camera}}^{i+1} = \text{InvRodrigues}(\text{base\_R}_{\text{hand}}) \]
11    \[ p_{\text{camera}}^{i+1} = (s_{\text{camera}}^{i+1}, r_{\text{camera}}^{i+1}) \]
12 end
13 return \( p_{\text{camera}}^{i+1} \)
14 end

left, right, up, down, inward, and outward regions. When \( s_{\text{palm}} \) reaches other regions except the home region, a corresponding directional unit vector \( u_d \) and distance weight \( l \) (remain the displacement in an appropriate range) are computed to update the position part of the target pose \( p_{\text{camera}}^{i+1} \). However, if \( s_{\text{palm}} \) stays in the home region, the manipulator switches into rotation mode, then the Euler angles captured by the gesture sensor are to be transformed into rotation matrix \( \text{lm\_R}_{\text{hand}} \), then this rotation matrix will be left multiplied by a rotation matrix from the gesture sensor to the robot base coordinate system \( \text{base\_R}_{\text{lm}} \). With the resulting rotation matrix \( \text{base\_R}_{\text{hand}} \), the inverse Rodrigues formula is used to transform it into the corresponding rotation part \( r_{\text{camera}}^{i+1} \) to update \( p_{\text{camera}}^{i+1} \).

During the process, the left hand’s gesture is used to store the point cloud data \( P \) 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, \( \{P_1, P_2, \ldots, P_n\} \), and their calculated camera pose, \( \{p_{\text{camera}}^1, p_{\text{camera}}^2, \ldots, p_{\text{camera}}^n\} \), 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 \( P \), the transformation from camera coordinate frame, \( F_{\text{camera}} \), to the base frame, \( F_{\text{robot}} \), is given by:

\[
^bF_{P} = {}^bT_c \cdot ^cF_{P}
\]

(1)

where \( ^bT_c \) is a fixed rigid transformation from the depth camera to robot obtained by hand-eye calibration [16], and \( ^bT_r \) 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 \( \{P_1, P_2, \ldots, P_n\} \), the entire point cloud data has been set in the base coordinate system, then they can be combined as a completed point cloud \( \{^bP_1 \cup ^bP_2 \cup \ldots \cup ^bP_n\} \) for the welding workpiece. In further improving the fitness, an ICP process setting the transformation from the depth camera to robot base frame, \( ^bT_c \), as its initial configuration is needed. Specifically, the \( i \)-th source point is defined as \( s_i = (s_{ix}, s_{iy}, s_{iz})^T \), its corresponding target point is represented by \( t_i = (t_{ix}, t_{iy}, t_{iz})^T \), and the unit normal vector at \( t_i \) can be denoted as \( n_i = (n_{ix}, n_{iy}, n_{iz}, 0)^T \). The goal of each ICP iteration is to obtain \( T_i \), that is:

\[
T_i = \arg \min_T \sum_i \left( (T \cdot s_i - t_i) \cdot n_i \right)^2
\]

(2)

where \( T_i \) and \( T_e \) are 4×4 3D target and initial transformation matrices. In its first iteration, \( ^bT_r \) is set as an initial transform matrix to accelerate the convergence. Usually, a rotation matrix \( R(\alpha, \beta, \gamma) \) and a translation matrix \( Trans(p_x, p_y, p_z) \) can be decomposed into a 3D transformation \( T \), which can be denoted as:

\[
T = Trans(p_x, p_y, p_z) \cdot R(\alpha, \beta, \gamma)
\]

(3)

Eq. 2 is a essentially a least-square optimization problem whose solution normally requires the determination of the six parameters \( \alpha, \beta, \gamma, p_x, p_y, \) and \( p_z \). However, in \( R(\alpha, \beta, \gamma) \), three of them \( (\alpha, \beta, \gamma) \) correspond to the arguments of nonlinear 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:

\[
\hat{R}(\alpha, \beta, \gamma) \approx \begin{bmatrix}
1 & -\gamma & \beta \\
\gamma & 1 & -\alpha \\
-\beta & \alpha & 1
\end{bmatrix}
\]

(4)

\[
\hat{T} = Trans(p_x, p_y, p_z) \cdot \hat{R}(\alpha, \beta, \gamma)
\]

(5)

\[
\hat{T}_i = \arg \min_T \sum_i \left( (T \cdot s_i - t_i) \cdot n_i \right)^2
\]

(6)

where \( (T \cdot s_i - t_i) \) can be written as a linear expression with six parameters \( (\alpha, \beta, \gamma, p_x, p_y, p_z) \). Then, \( N \) pairs of point correspondences can be arranged as a linear matrix expression as follows:

\[
\min_T \sum_i \left( (T \cdot s_i - t_i) \cdot n_i \right)^2 = \min_x |Ax - b|^2
\]

(7)

Thus, \( \hat{T}_i \) can be obtained by solving:

\[
x_r = \arg \min_x |Ax - b|^2
\]

(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]
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 reconstructed workpiece model varies, few points of a neighborhood exist within a fixed range. The establishment of a minimum point will provide a rational neighborhood to its neighbor. Gradient $I_{o}$, defined as the difference between $I_{p}$ 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 $H$, which is calculated by:

$$F(x_{o}, y_{o}, z_{o}) = \sum_{\delta x, \delta y, \delta z} \left[ I_{x_{o}+\delta x, y_{o}+\delta y, z_{o}+\delta z} - I_{x_{o}, y_{o}} \right]^{2}$$

$$= \sum_{\delta x, \delta y, \delta z} \left[ \delta x \cdot I_{x} + \delta y \cdot I_{y} + \delta z \cdot I_{z} \right]^{2} + O(\delta x^2, \delta y^2, \delta z^2)$$

$$\approx (\delta x, \delta y, \delta z)^{T}H(\delta x, \delta y, \delta z)$$

where the distribution of the gradients for the local neighborhood is depicted by the eigenvalues of $H$. Therefore, the intensity of gradient changes can be detected by comparing the difference of eigenvalues. For example, a high eigenvalue reveals a large shift gradient changes can be detected by comparing the difference of eigenvalues. For example, a high eigenvalue reveals a large shift.

Most importantly, this ratio calculation scheme does not rely on a multiplication of a small offset $(\delta x, \delta y, \delta z)$ and a symmetric matrix $H$, which is calculated by:

$$I_{k} \approx \max_{i \in N} \left[ (I_{i} - I_{o}) \cdot \frac{d_{i,o}}{\delta k} \right], k = x, y, z$$

where $\delta x, \delta y, \delta z$ 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 $H$, which is calculated by:

$$F(x_{o}, y_{o}, z_{o}) = \sum_{\delta x, \delta y, \delta z} \left[ I_{x_{o}+\delta x, y_{o}+\delta y, z_{o}+\delta z} - I_{x_{o}, y_{o}} \right]^{2}$$

$$= \sum_{\delta x, \delta y, \delta z} \left[ \delta x \cdot I_{x} + \delta y \cdot I_{y} + \delta z \cdot I_{z} \right]^{2} + O(\delta x^2, \delta y^2, \delta z^2)$$

$$\approx (\delta x, \delta y, \delta z)^{T}H(\delta x, \delta y, \delta z)$$

where the distribution of the gradients for the local neighborhood is depicted by the eigenvalues of $H$. 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 $\lambda_{1} = \alpha \cdot \lambda_{3}$ and $\lambda_{2} = \beta \cdot \lambda_{3}$ ($\alpha \geq \beta \geq 1$) to represent the three sorted eigenvalues ($\lambda_{1} \geq \lambda_{2} \geq \lambda_{3}$). Subsequently, the ratio between trace $\text{Tr}(H)$ and determinant $\text{Det}(H)$ can be calculated using 14. As the ratio increases with larger $\alpha$ and $\beta$, we can set a threshold $t$ to guarantee the existence of
large eigenvalues:
\[
\text{Tr}(\mathbf{H})^3 = \frac{(\lambda_1 + \lambda_2 + \lambda_3)^3}{\lambda_1 \lambda_2 \lambda_3} = \frac{(\alpha + \beta + 1)^3}{\alpha \beta} \geq t
\]  
(14)
\[
G(o, i) = \frac{1}{\delta^2} e^{-\frac{\|p_o - p_i\|^2}{2\delta^2}}, \quad q_i \in \mathcal{N}
\]  
(15)

Then, Eq. (15) is used to smoothen \( \mathbf{H} \) and ensure \( \text{Det}(\mathbf{H}) \) is a non-zero, in which \( \sigma \) 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 \( \{p_1(e), p_2(e), \ldots, p_n(e)\} \) is generated.

E. Welding Path Optimization

On the basis of the extracted edge points, \( p_i(w) = p_i(e) + d \times n_i(p_i(e)) \) 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 \( \{p_1(w), p_2(w), \ldots, p_n(w)\} \) 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 \( \rho \) points is interpolated between \( p_i(w) \) and \( p_{i+1}(w) \) and the \( j \)-th interpolation point is defined as:
\[
p_j(w) = p_i(w) + j \cdot \frac{p_{i+1}(w) - p_i(w)}{\rho}
\]  
(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:
\[
\mathbf{v}_j(w) = \sin((1-\alpha)\theta)\mathbf{v}_i(w) + \sin(\alpha\theta)\mathbf{v}_{i+1}(w) \overline{\sin \theta}
\]  
(17)

where \( \alpha = j/\rho \) and \( \theta = \arccos(\mathbf{v}_i(w) \cdot \mathbf{v}_{i+1}(w)) \). 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
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’ 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:

\[ S(M) = \frac{\|P_m\|}{\|P_t\|}, \left\{ \left| t_i - Mm_i \right\| < d \right\} \]  

where \( t_i \) and \( m_i \) are the target point and the model point, respectively, and 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:

\[ F_1 = \frac{3}{a_{xy}^{-1} + a_{xz}^{-1} + a_{yz}^{-1}} \]  

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 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.
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 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

\[
(p_i - a)^2 + (q_i - b)^2 = r^2, q_i > b
\]

\[
-(p_i - a)(z_i - q_i) = -1
\]

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:

\[
RMSE(m, g) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (m_i - g_i)^2}
\]
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 downsampling 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 are 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, welding experiments will be conducted with our proposed system. Additionally, we may introduce other new interaction techniques to assist robotic welding tasks.