

Path Planning over Point Cloud Models with Hand Gesture Guided Scanning for Robotic Arc Welding

Peng Zhou, Rui Peng, Maggie Xu, Victor Wu and David Navarro-Alarcon

Abstract—This paper 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.

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. With the 3D reconstruction, 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 validate the proposed methods, we conduct detailed experiments on workpieces with different joint types.

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,

All authors are with The Hong Kong Polytechnic University, Kowloon, Hong Kong. Corresponding author e-mail: dna@ieee.org

This work is supported in part by the Research Grants Council of HK (grant 14203917), in part by the Chinese National Engineering Research Centre for Steel Construction Hong Kong Branch (grant BBV8), and in part by PolyU under grant YBYT and ZZHJ.

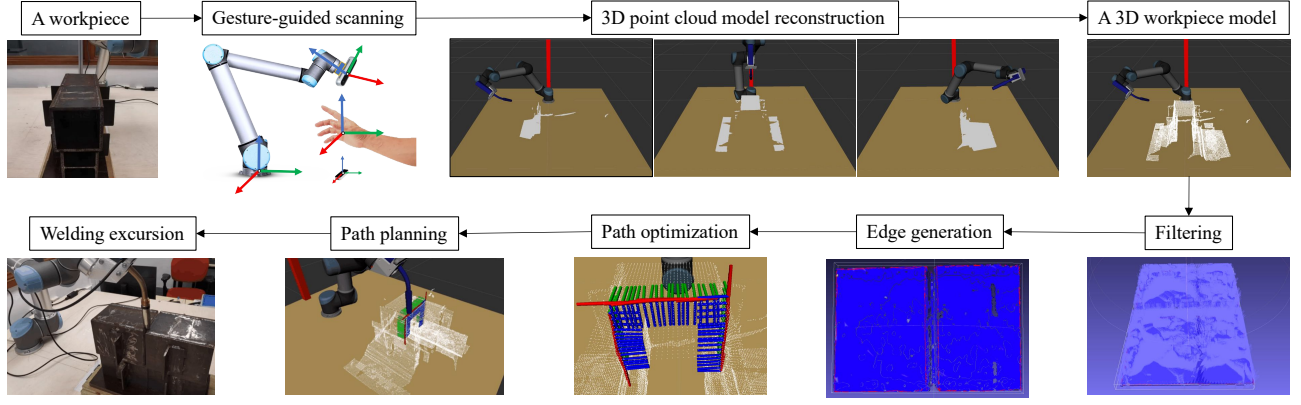


Fig. 1. Overview of the automatic robotic welding pipeline. Given a workpiece, a gesture-guided scanning is executed to merge captured point clouds into a completed 3D model, which is further improved with the bilateral filter. Subsequently, on basis of the edge points detected using an edge intensity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

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.

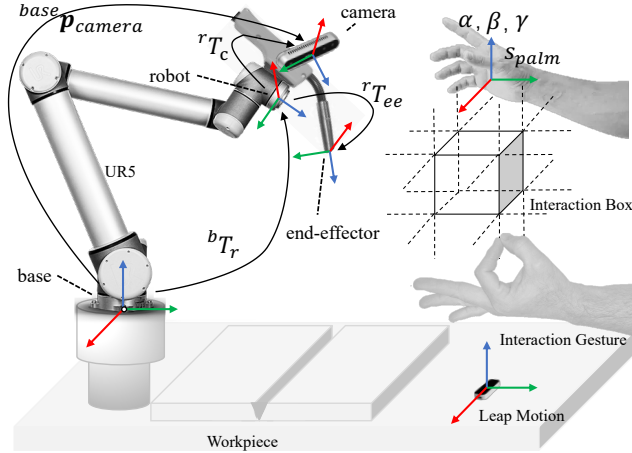


Fig. 2. Conceptual representation of the intuitively interactive scanning with hand gestures, in which an interaction box for right hand is built to define different mapping rules for robotic motions and “ok” gesture for left hand to capture the point cloud data.

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, 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. According to the relative position with respect to the interaction box, the entire interaction space is separated into seven regions, namely, home, left, right, up, down, inward, and outward regions. Therefore, 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 capturing process in appropriate camera perspectives.

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      $s_{camera}^{t+1} += u_d \cdot l$ 
5      $p_{camera}^{t+1} = (s_{camera}^{t+1}, r_{camera}^t)$ 
6   end
7 else
8      ${}^{lm}R_{hand} = \text{Transform}(\alpha, \beta, \gamma)$ 
9      ${}^{base}R_{hand} = {}^{base}R_{lm} \cdot {}^{lm}R_{hand}$ 
10     $r_{camera}^{t+1} = \text{InvRodrigues}({}^{base}R_{hand})$ 
11     $p_{camera}^{t+1} = (s_{camera}^t, r_{camera}^{t+1})$ 
12  end
13  return  $p_{camera}^{t+1}$ 
14 end

```

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 (α, β, γ) , and the current pose with respect to the camera ${}^{base}p_{camera}^t$ as the inputs. The resulting desired output is target pose ${}^{base}p_{camera}^{t+1}$ 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. When s_{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 ${}^{base}p_{camera}^{t+1}$. However, if s_{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 ${}^{lm}\mathbf{R}_{hand}$, 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.

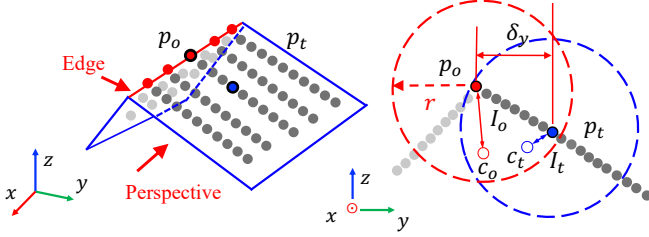


Fig. 3. Conceptual representation of the edge intensity and gradient along the Y-axis for current point p_o , in which c_o represents the local geometric centroid for p_o and edge intensity I_o with respect to p_o is higher than I_t with respect to its neighbor p_t . Gradient I_{oy} is defined as the difference between I_o and I_t divided by the coordinate difference along the Y-axis.

C. Welding Seam Extraction

To remove noise from the point cloud data without removing vital edge information, an extension of bilateral filter [15] 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:

$$\mathcal{N}_r(p) = \{q \in \mathcal{P} \mid \|q - p\|_2 < r \cup |\mathcal{N}| \geq m\} \quad (1)$$

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. 3, 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:

$$I_o = \frac{1}{r} \left\| p_o - \frac{1}{n} \sum_{i=0}^{i=n} q_i \right\| \quad (2)$$

where $\frac{1}{n} \sum_{i=0}^{i=n} q_i$ 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. 3 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 .

On the basis of the 3D gradient operators [16], we define the 3D gradient decompositions for a point p_o as:

$$I_k \approx \max_{i \in \mathcal{N}} \left[(I_i - I_o) \cdot \frac{d_{i,o}}{\delta k} \right], k = x, y, z \quad (3)$$

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. 3 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 [17] 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:

$$\begin{aligned} F(x_o, y_o, z_o) &= \sum_{\delta x, \delta y, \delta z} [I_{x_o + \delta x, y_o + \delta y, z_o + \delta z} - I_{x, y, z}]^2 \\ &= \sum_{\delta x, \delta y, \delta z} \left[\begin{array}{c} \delta x \cdot I_x + \delta y \cdot I_y + \delta z \cdot I_z \\ + O(\delta x^2, \delta y^2, \delta z^2) \end{array} \right]^2 \\ &\approx (\delta x, \delta y, \delta z) \mathbf{H} (\delta x, \delta y, \delta z)^\top \end{aligned} \quad (4)$$

$$\mathbf{H} = \begin{bmatrix} I_x^2 & I_x I_y & I_x I_z \\ I_y I_x & I_y^2 & I_y I_z \\ I_z I_x & I_z I_y & I_z^2 \end{bmatrix} \quad (5)$$

Inspired by approach in [17], a set of edge points $\{p_1^{(e)}, p_2^{(e)}, \dots, p_n^{(e)}\}$ is generated.

III. RESULTS

A. Gesture-Guide Scanning

To validate the performance of scanning the workpiece with bare hands, the experiment uses right hand palm position to guide the depth sensor complete 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 5, 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 5cm 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. The maximum average errors at three points are 4.3cm, 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.

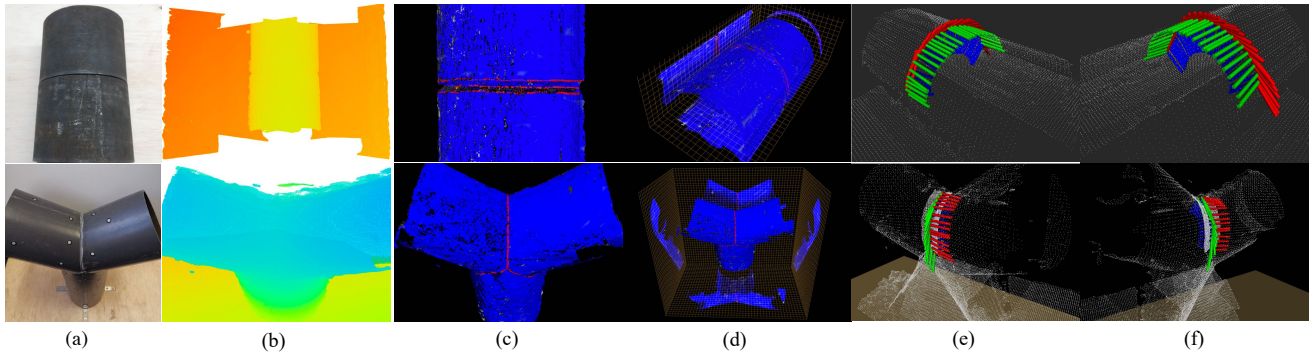


Fig. 4. Results of different processes in the proposed system. (a) Two actual welding metal workpieces, cylinder and Y-shape. (b) Reconstructed models of workpieces with a depth map. (c) Welding seam extracted with the edge intensity gradient-based algorithm. (d) Rasterized views of extracted edge points. (e, f) 6-DOF welding path after interpolation.

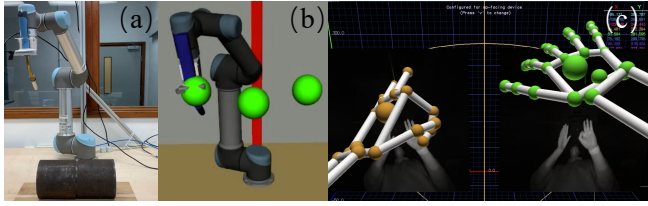


Fig. 5. (a) Real environment. (b) Three waypoints in Rviz. (c) Hand gestures in Leap Motion visualizer.

B. Welding Path Detection

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

$$F_1 = \frac{3}{a_{xy}^{-1} + a_{xz}^{-1} + a_{yz}^{-1}} \quad (6)$$

where a_{xy}^{-1} , a_{xz}^{-1} and a_{yz}^{-1} denote the accuracy calculated for the xy , xz , and yz planes after rasterization. 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 results of two workpieces during the entire process in the proposed system is shown in Fig. 4. For more detailed results, please refer to [18].

IV. CONCLUSION

This article presents a sensor-guided welding robotic system [19]. 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. 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.

REFERENCES

- [1] Y. Xu, H. Yu, *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, S. Deng, *et al.*, "An initial point alignment and seam-tracking system for narrow weld," *IEEE Trans. Industr. Inform.*, vol. 16, no. 2, pp. 877–886, 2019.
- [3] C. Diao, J. Ding, S. Williams, Y. Zhao, *et al.*, "A passive imaging system for geometry measurement for the plasma arc welding process," *IEEE Trans. Ind. Electron.*, vol. 64, no. 9, pp. 7201–7209, 2017.
- [4] X. Li, X. Li, *et al.*, "Automatic welding seam tracking and identification," *IEEE Trans. Ind. Electron.*, vol. 64, no. 9, pp. 7261–7271, 2017.
- [5] H. N. M. Shah *et al.*, "Review paper 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, P. Phairatt, *et al.*, "Feature extraction and tracking of a weld joint for adaptive robotic welding," in *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, *et al.*, "Edge and corner detection for unorganized 3d point clouds with application to robotic welding," in *IEEE/RSJ Int. Conf. on Robots and Intelligent Systems*, 2018, pp. 7350–7355.
- [12] V. Patil, I. Patil, *et al.*, "Extraction of weld seam in 3d point clouds for real time welding using 5 dof robotic arm," in *Int. Conf. Cont., Auto. and 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 *Chinese Control and Decision 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] J. Digne and C. De Franchis, "The bilateral filter for point clouds," *Image Processing On Line*, vol. 7, pp. 278–287, 2017.
- [16] Y. Zhang, "Quantitative study of 3d gradient operators," *Image and vision computing*, vol. 11, no. 10, pp. 611–622, 1993.
- [17] C. G. Harris *et al.*, "A combined corner and edge detector," in *Alvey vision conference*, vol. 15, no. 50, 1988, pp. 10–5244.
- [18] P. Zhou, R. Peng, M. Xu, V. Wu, and D. Navarro-Alarcon, "Path planning with automatic seam extraction over point cloud models for robotic arc welding," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5002–5009, 2021.
- [19] A. Cherubini and D. Navarro-Alarcon, "Sensor-based control for human-robot collaboration: Fundamentals, challenges and opportunities," *Front Neurobotics*, vol. 14, p. 113, 2021.