A Point Cloud Based Method for Automatic Groove Detection and Trajectory Planning of Robotic Arc Welding Tasks

Rui Peng, David Navarro-Alarcon, Victor Wu, Wen Yang

***Abstract*— In this paper, we propose a point cloud based method driven by an RGB-D camera for robotic arc welding. The method consists of two parts: groove detection algorithm and 3D welding trajectory planning. Since actual welding scenes could be represented in format of 3D point cloud generated by an RGB-D camera, the welding groove detection algorithm purely based on 3D point cloud focuses on geometrical feature of welding groove. The detection algorithm is capable of well adapting to general welding workpieces with different types of welding groove. Meanwhile, the 3D welding trajectory involving 6-DOF poses of the welding groove for robotic manipulator motion is generated. With acceptable error of the trajectory planning, the robotic manipulator could drive the welding torch to follow the trajectory and execute welding tasks. Also details of the integrated robot system are presented in this paper. Experimental results show application feasibility of the proposed method.**

I. INTRODUCTION

Nowadays industrial robotic manipulators are implemented extensively in many factories around the world. It is quite time-consuming to program these manipulators manually by experienced robot operators which is the pattern of teach-playback. More specifically, in order to manipulate a robotic arm for welding tasks, a human operator needs to set each path point of welding trajectories with precise three-dimensional position and three-dimensional orientation in advance. This sort of conventional robotic use lacking of efficiency has been already unable to cope with growing welding demand of construction industry.

To improve the weakness, it is encouraging to employ various sensors with the manipulators for more intelligent welding tasks. Prevailing perceiving systems involve vision sensors [1]–[3], RGB-D sensors [4]–[6], ultrasonic sensors [7], sound sensors [8], infrared sensors [9] and laser sensors [10]. During actual robotic welding tasks under complex and unpredictable situation, it is quite essential to plan a precise trajectory [11] of the welding groove between two workpieces to achieve acceptable welding quality. Recently, 2D imaging processing methods have been a hot-point and a useful technology to assist robotic welding in various industrial environment [12]. With respect to common welding tasks, 2D vision sensors capture one frame image of workpieces. L. Nele · E et al. [13] established an image acquisition system using a CCD video camera for real-time weld seam tracking.

However, 2D imaging processing algorithms relying on color information cannot address dramatic environmental brightness variation particularly in welding fields. Researchers have been preferring to 3D vision sensors due to obtaining abundant information of welding fields. Also with advancement of Point Cloud Library (PCL) [14] specially designed for 3D point cloud processing, it is possible to extract and locate the target region in surface point cloud of welding workpieces. One type of 3D sensors, the stereo camera is used to realize welding seam reconstruction and tracking [15], achieving good planning accuracy. Compared with state-of-the-art algorithms on RGB-D semantic segmentation and 3D washer models from ShapeNet dataset, an unorganized point cloud based edge and corner detection algorithm [16] is proved for applicable robotic welding. With the emergence of inexpensive RGB-D sensors such as Microsoft Kinect, Intel RealSense and Apple PrimeSense [17], some approaches fusing RGB images and depth information have been developed [18]. Through integrating an RGB-D sensor into the robot system [19], the controller is able to be implemented in dynamic welding environment. Due to RGB images used to detect the weld groove and point cloud used to obtain the 3D position of weld groove, Li et al. [20] proposed a welding groove recognizing approach using an RGB-D camera.

One of the main problems in above research works based on 3D sensors is that their experimental results lack of testing different types of welding workpieces sufficiently. Inspired by a research work [21], the proposed method in this paper focuses on welding groove detection and 3D motion trajectory planning. Experimental results involve three aspects: runtime efficiency, groove detection accuracy and trajectory execution. The performance of the system testing four general types of workpieces in experiment prove its feasibility in actual welding application. Besides, the demonstration video is online: https://vimeo.com/371773986. And main contributions of this paper are following:

1) An integrated intelligent robot system with characteristic of light-weight is designed for executing welding tasks efficiently.
2) A point cloud processing based welding groove detection algorithm for unpredictable workpieces is proposed.
3) Automatic 3D welding trajectory planning method is implemented on a 6-DOF robotic arm.

This work is supported in part by the Research Grants Council of Hong Kong under grant 142039/17E and in part by the CNERC under grant BBV8.

RP, DNA, and VW are with The Hong Kong Polytechnic University, Department of Mechanical Engineering, Kowloon, Hong Kong. Corresponding author: dna@ieee.org

WY is with the Wuhan University, Hubei, People's Republic of China.

II. ROBOT SYSTEM IMPLEMENTATION

The integrated robot system contains a robotic manipulator (Universal Robot 3), an RGB-D camera (Intel Realsense D415) and a welding torch. The end-effector part and the whole experimental platform are shown as Fig. 1. The welding torch is installed on the end-wrist of the robotic manipulator tightly by supportive metal structure for stable welding execution. Also, the RGB-D camera is physically connected to the welding torch instead of being installed at a fixed position within the welding platform. The design of the RGB-D camera moving with the welding torch is able to cope with unpredictable welding situation flexibly. And it is simple to run hand-eye calibration for obtaining the quite accurate transformation matrix between the camera coordinate and UR3 base coordinate.

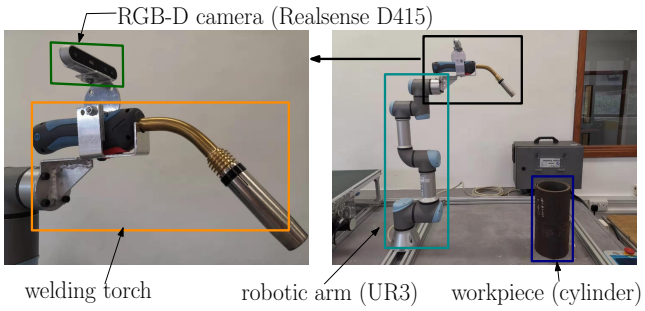

Fig. 1. (on left) End-effector of the proposed welding robot system is composed of an RGB-D camera and a welding torch. The camera is situated above the welding torch. (on right) Experimental welding platform.

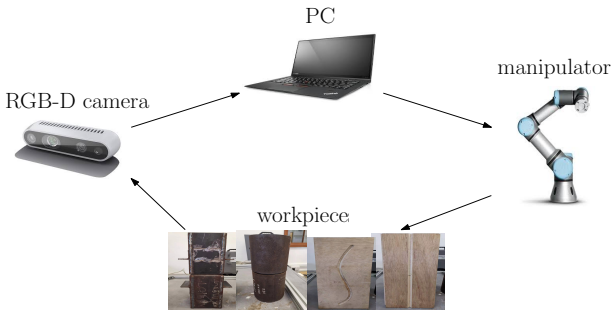

Fig. 2. Workflow of the proposed robot system: (1) the RGB-D camera captures surface point cloud of workpieces. (2) the welding groove detection algorithm locates the welding groove region in the input point cloud. (3) the trajectory planning method processes the groove point set and outputs a 3D welding trajectory. (4) the manipulator automatically executes the welding motion while tracking the generated trajectory.

The system running procedure is shown in Fig. 2 in which the PC with ubuntu 16.04 is Lenovo-Thinkpad whose CPU uses Intel i5. ROS [22], an open source robot operating system is used to build up software framework which makes it convenient to develop algorithm modules. The welding groove detection totally relies on PCL to process point cloud and extract geometric feature. Moveit [23] package as one part of the software system is utilized to control the UR3 robotic arm with motion planning and collision avoidance.

III. WELDING GROOVE DETECTION

The welding groove detection algorithm extracts the groove region by computing geometric feature of input point cloud which represents surface profile of welding workpieces. The geometric feature is defined as surface variation which means degree of surface slope change. Compared with flat and smooth regions, the groove region has higher surface variation, shown as Fig. 3.

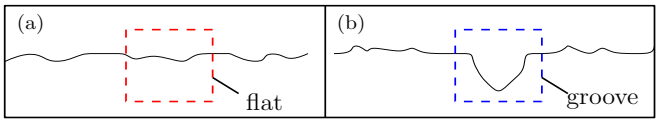

Fig. 3. Geometric feature diagram of surface. (a) flat region. (b) groove region.

In order to describes surface variation mathematically, a novel 3D feature histogram in the algorithm is designed. Furthermore, a descriptor called Surface Variation Descriptor is computed for each point of the input point cloud. Thus the descriptor is used to distinguish between the groove region and other regions.

A. Input Point Cloud Preprocessing

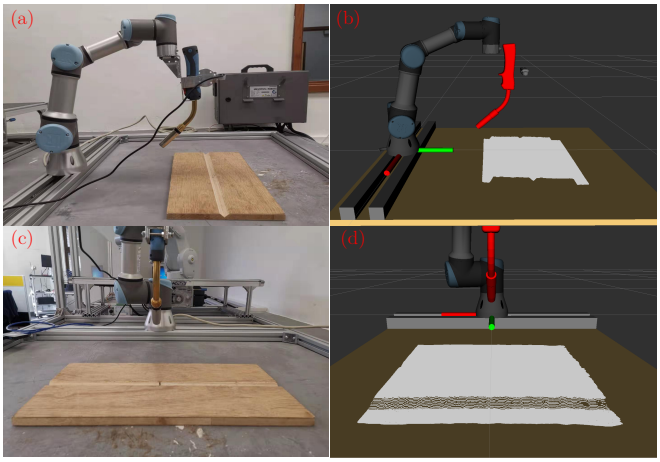

Fig. 4. RGB-D camera captures surface point cloud of the workpieces with straight-line welding groove. (a) robotic manipulator moves to a proper pose so that the camera could capture complete point cloud. (b) the surface point cloud is shown in simulation (Rviz). (c) the real workpieces in side view. (d) the raw point cloud is organized.

In the first beginning, the RGB-D camera moves to an appropriate pose by the robotic arm and captures one frame of raw point cloud covering the whole surface of workpieces, shown as Fig. 4. Then the raw point cloud is taken as input of the welding groove detection algorithm.

In general there is noise data in the raw point cloud owing to hardware factors of the camera. Thus the point cloud needs smoothing before next step. PCL provides a Moving Least Squares (MLS) surface reconstruction method to smooth point cloud surface and reduce noisy data. Fig. 5 shows before-after result of smoothing processing for point cloud of the testing welding workpieces in Fig. 4. It clearly

demonstrates that after smoothing the point cloud has more even and gentle surface.

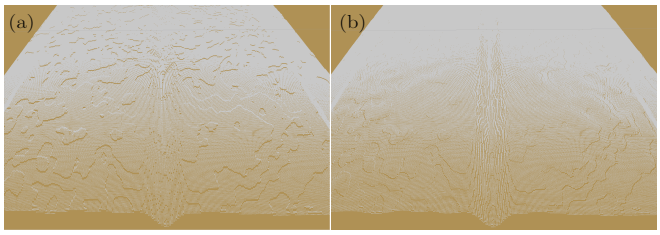

Fig. 5. Raw point cloud smoothing. (a) point cloud before smoothing has too many noisy regions. (b) point cloud after smoothing has more even surface.

Surface normal computation is a fundamental part for welding groove detection. Also PCL offers mathematical method to estimate surface normal of each point. Theoretically, given a cluster of point cloud, computing one point normal actually is a problem of estimating a normal of a plane tangent to the point cloud surface. Of course, the plane must pass through this point. Simply put, one point's neighbor (a sphere with a constant radius and the point is the sphere's center) is a small cluster of point cloud which could fit a plane. Therefore normal of this plane is regarded as normal of the point.

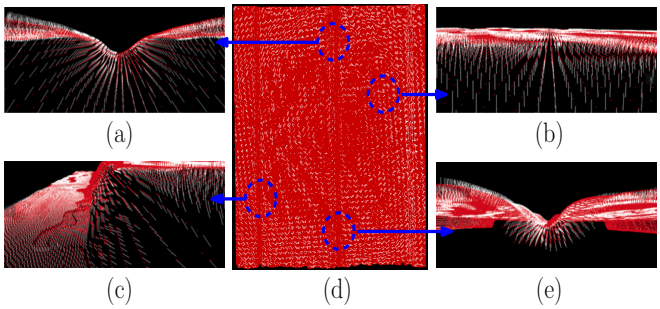

Fig. 6. Surface normal map. Each white arrow represents the normal of each point. (a) groove region. (b) flat region. (c) edge region. (d) surface point cloud. (e) groove region.

Then normalized map of surface normal shown in Fig. 6 for the after-smoothing point cloud is obtained by iterating every point with least-square plane fitting. Moreover, each point normal $\vec{u_i}$ shown as a white arrow in Fig. 6 is a unit vector defined as:

$$\vec{u_i} = [x_i, y_i, z_i]^T, \sqrt{(x_i)^2 + (y_i)^2 + (z_i)^2} = 1 \qquad (1)$$

where i represents the i-th point in organized point cloud.

Due to an inspiring research work [24] for object recognition, a descriptive method called Groove Feature Histogram (hereinafter referred to as GFH) is designed in this section to quantify surface variation degree. The GFH computed for each point of the point cloud has two types: local GFH extracting the feature of local surface variation and global GFH considering the whole point cloud based surface variation. Furthermore, through local GFH and global GFH of one point, Surface Variation Descriptor of this point mentioned in Sec. III is generated.

As previous discussed, in mathematics one surface normal in 3D space is expressed by a unit vector consists of three parameters. Besides, each normal has its own main direction.

B. Local GFH

Since one point's neighbor is defined as a sphere (the point is its central point) with a constant radius r. Within this neighbor, the kdTree search method is used to find other neighbor points (Euclidean distance to the central point is not more than r). Points of the neighbor is regarded as a individual point set where the normal of each point could be obtained from the surface normal map (Fig. 6). Then the central point is paired with every other neighbor point and every pair has two unit vectors which form one included angle. Fig. 7 shows one point's neighbor with one pair.

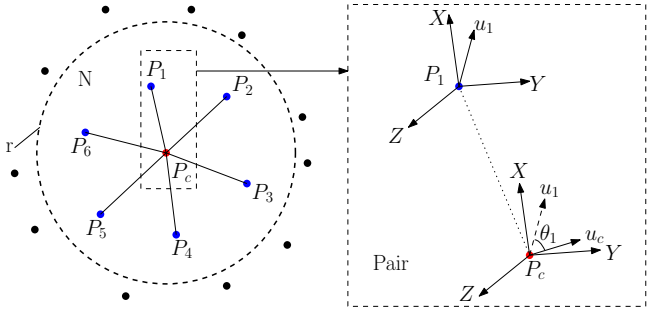

Fig. 7. (on left) The neighbor N of one point P is defined as a sphere whose radius is r. Within the neighbor N, the red point is the central point P_c of the sphere and the blue points are neighbor points. (on right) The central point P_c with its normal u_c is paired to one neighbor point P_1 with its normal u_1. θ_1 is the included angle between u_c and u_1.

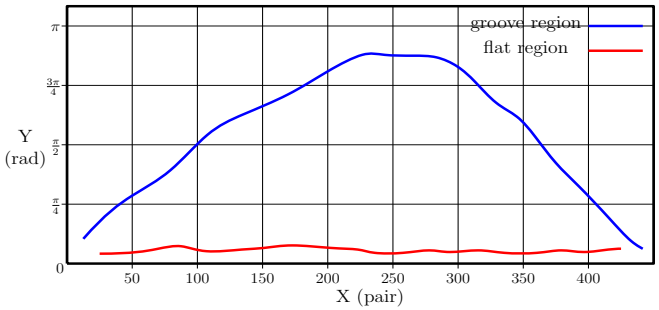

Fig. 8. Local groove feature histogram. The groove region has higher variation than the flat region.

Generally since 3D position of all the points are relevant to the manipulator base coordinate system through rigid transformation, all the point normals are based on the same coordinate XYZ as shown in Fig. 7 (right). Furthermore, the included angle θ_j of the j-th pair within the center point's neighbor is computed as:

$$\theta_j = \arccos \frac{\vec{u_c} \cdot \vec{u_j}}{|\vec{u_c}| \cdot |\vec{u_j}|} \qquad (2)$$

where $\vec{u_c}$ is normal of the center point and $\vec{u_j}$ is normal of the j-th neighbor point paired to the center point. By using Equation (2) for iterating every pair of the neighbor, all the computation results are arranged as a set ($\Theta = \{\theta_1, \theta_2, ..., \theta_n\}$, n is number of the neighbor points), which builds up local GFH of the center point. Fig. 8 shows local GFH of one point from the groove region and one point from flat regions.

C. Global GFH

As for global GFH, it needs to take the whole point cloud into account. Therefore unit benchmark normal $\vec{u_b}$ representing main direction of the whole point cloud is defined as:

$$\vec{U_b} = \sum_{i=0}^{n} \vec{u_i}, \vec{u_b} = \frac{\vec{U_b}}{\left|\vec{U_b}\right|} \tag{3}$$

where $\vec{u_i}$ is the i-th point normal of the point cloud.

Still discussing neighbor of the previous point (the central point P_c in Fig. 7), each point normal in the neighbor is paired to the benchmark normal $\vec{u_b}$, shown as Fig. 9.

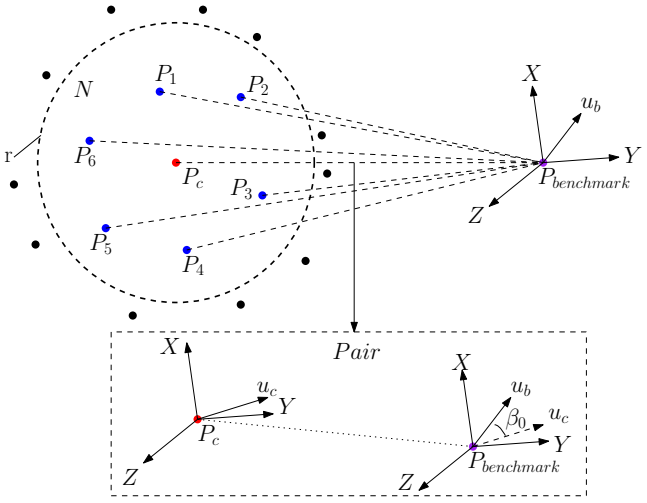

Fig. 9. Every point normal of the neighbor is paired to the benchmark normal. For example of one pair (the central point normal $\vec{u_c}$ and the benchmark normal $\vec{u_b}$), β_0 is the included angle between $\vec{u_c}$ and $\vec{u_b}$.

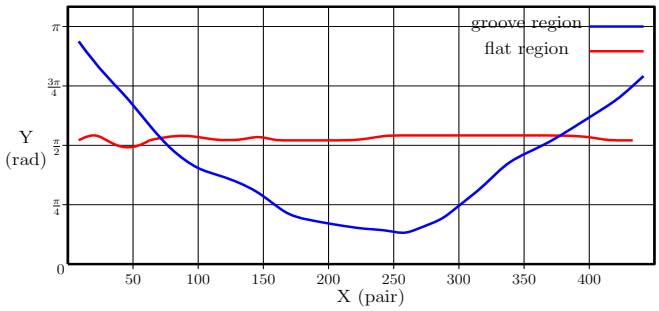

Fig. 10. Global groove feature histogram. The groove region has higher variation than the flat region.

In the same principle as local GFH, the k-th included angle β_k of the neighbor is computed as:

$$\beta_k = \arccos \frac{\vec{u_b} \cdot \vec{u_k}}{|\vec{u_b}| \cdot |\vec{u_k}|} \tag{4}$$

Therefore all the resultant included angles are put into one set ($B = \{\beta_1, \beta_2, ..., \beta_n\}$) which builds up global GFH of the central point. Fig. 10 shows global GFH of one point from the groove region and one point from flat regions.

D. Surface Variation Descriptor

According to local GFH III-B and global GFH III-C of one point, the surface variation descriptor D_i for i-th point P_i of the point cloud is defined as:

$$D_i = \sqrt{(\sigma_i^l)^2 + (\sigma_i^g)^2} \tag{5}$$

where σ_i^l is variation of local histogram of P_i and σ_i^g is variation of global histogram of P_i. Through computing D_i of every point, a map representing surface variation degree of the whole point cloud is obtained and shown as Fig. 11. It intuitively shows degree of surface variation on the whole point cloud, although there are noisy places. And the blue regions such as groove region and edges has high variation and white regions are almost flat.

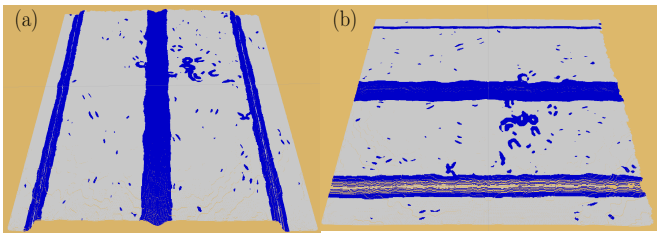

Fig. 11. Surface Variation Map. (a) Front view. (b) Side view.

By analyzing the surface variation map, points within the groove region are centralized and almost connected tightly. Thus it is simple to screen out the groove region and delete all the irrelevant points. Ultimately, only groove point set is left and groove detection result (blue region) is shown in Fig. 12. In next step the groove point set would be used to generate welding trajectory for the robotic arm.

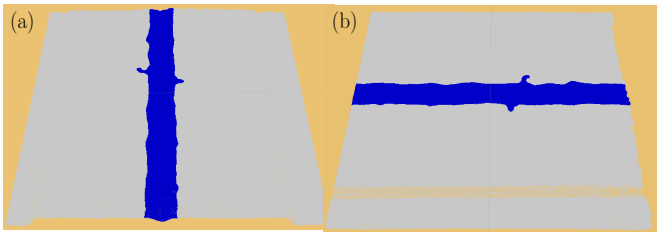

Fig. 12. Groove Detection Result. (a) Front view. (b) Side view

IV. WELDING TRAJECTORY PLANNING

The welding trajectory planning is totally based on the groove point set obtained by the groove detection algorithm as shown in Fig. 12. Here the point set is segmented into a series of consecutive regions (blue, green and red for distinction), shown as Fig. 13. And each segment region generates one way point. All the way points together form the final trajectory.

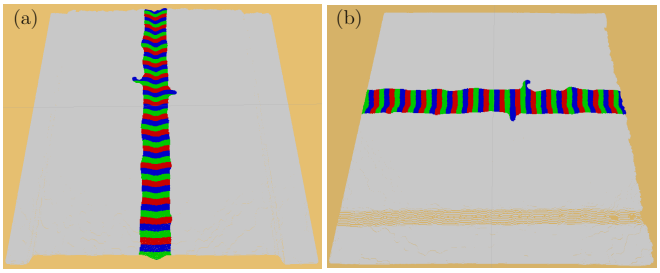

Fig. 13. Groove Point Set Segmentation. (a) Front view of segmented groove region. (b) Side view of segmented groove region.

The sum of space distance from the way point of each segmented region to every point of the same segmented region should be as short as possible. And this problem could be defined as a function for each segmented region:

$$\vec{I_i^s} = P^w - P_i^s, f = argmin \sum_{i=0}^{m} \left|\vec{I_i^s}\right| \tag{6}$$

where P^w is the unknown way point and P_i^s is the i-th point of one segmented region. When f gets to its minimum by gradient descent method, P^w is obtained as the way point.

After iterating every segmented region of the groove point set by Equation (6), the final welding trajectory planning result is shown as Fig. 14.

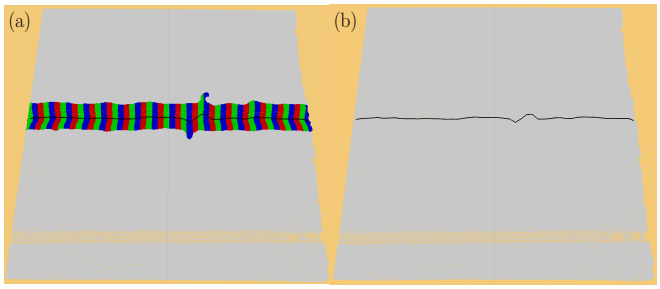

Fig. 14. Welding trajectory planning. (a) Generated welding trajectory inside the segmented groove point set. (b) The trajectory inside the raw point cloud.

And the three dimensional orientation of each way point of the generated trajectory is represented as a unit vector $\vec{o_w}$, defined as:

$$\vec{O_w} = \sum_{i=0}^{m} \vec{u_i^s}, \vec{o_w} = \frac{\vec{O_w}}{\left|\vec{O_w}\right|} \tag{7}$$

where $\vec{u_i^s}$ is the i-th point normal of the segmented region.

V. RESULTS

An open welding environment in Fig. 1 with four types of welding workpieces is prepared for experiment. Fig. 15 presents experimental welding workpieces which are straight-line, curve-line, box and cylinder type respectively.

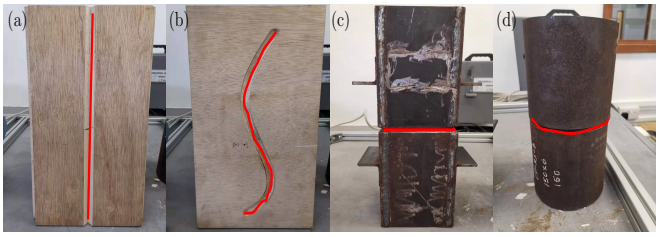

Fig. 15. Experimental welding workpieces with their welding groove colored by red. According to shape of welding groove, the workpiece is defined as: (a) Straight-line. (b) Curve-line. (c) Box. (d) Cylinder.

In order to objectively evaluate performance of the proposed method, three vital elements are considered:

1) The whole processing runtime from inputting raw point cloud to generating motion trajectory.
2) Overlapping rate of detected groove region and actual groove region.
3) Disparity between the generated trajectory and standard welding trajectory.

The processing runtime as an essential factor could evaluate efficiency of the proposed robot system for automatic welding. The overlapping rate is used to illustrate accuracy of the welding groove detection algorithm. The disparity is presented by actual automatic welding execution. Thus focusing on three elements above, in experiment each workpiece shown in 15 is tested. In fact, although the RGB-D camera is well calibrated, the error of measurement of depth still exists.

A. Processing Runtime Results

To measure processing runtime of the proposed method for surface point cloud of one workpiece, the function "clock()" of C++ is used to capture system start time and end time of the method. Due to running performance of the PC, for each workpiece we test ten times and record every runtime from inputting raw point cloud to generating motion trajectory, shown as Table I.

B. Groove Detection Results

To evaluate accuracy of the welding groove detection algorithm, the result of the algorithm needs to be compared with the ground truth (actual groove region defined by the authors). Inspired by the concept of IoU (Intersection over Union) in 2D image processing, the 3D overlapping rate λ of detected groove region and actual groove region is introduced:

$$\lambda = \frac{n_{overlap}}{n_d + n_{gt} - n_{overlap}} \tag{8}$$

where n_d is number of points of detected groove region, n_{gt} is number of points of actual groove region and $n_{overlap}$ is

TABLE I

RESULTS OF PROCESSING RUNTIME FOR EACH WORKPIECE

Types	P_{num}	t_1(s)	t_2(s)	t_3(s)	t_4(s)	t_5(s)
Straight-line	265800	14.01	13.93	14.15	14.16	14.06
Curve-line	266497	14.05	14.39	13.91	14.28	14.14
Box	127031	7.61	7.44	7.45	7.46	7.53
Cylinder	121429	6.11	6.05	6.03	6.10	5.98
	t_6(s)	t_7(s)	t_8(s)	t_9(s)	t_{10}(s)	t_{mean}(s)
	14.02	14.05	14.10	14.21	14.23	14.09
	13.92	14.18	14.08	14.05	13.82	14.08
	7.59	7.49	7.52	7.49	7.60	7.51
	6.07	6.20	5.97	6.02	6.03	6.05

P_{num} = Number of points. t_{mean} = Average time of runtime.

number of points of overlapping region between the detected groove region and the actual groove region.

The welding groove detection process for each workpiece in 15 is shown in Fig. 16. And detection accuracy (defined by 3D overlapping rate) results are presented in Table II.

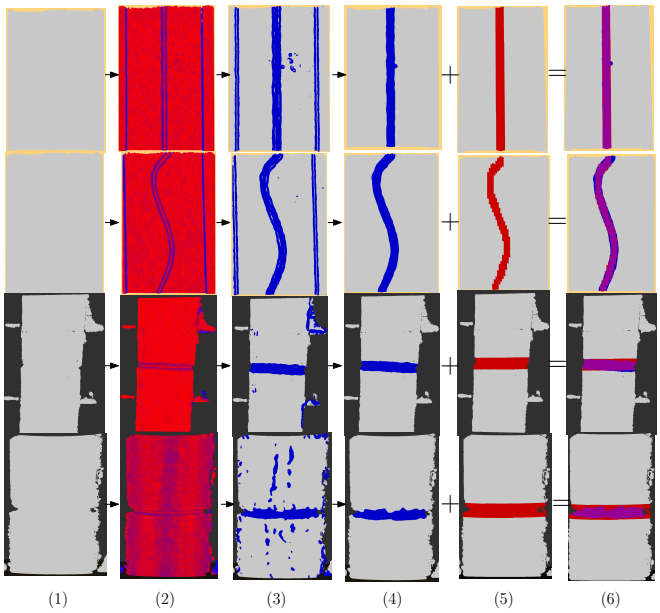

Fig. 16. Welding groove detection workflow. (1) Input raw point cloud of each workpiece. (2) Global normal map. (3) Surface variation map by groove feature histogram (GFH). (4) Filtering results (detected groove region). (5) Ground truth of groove region in raw point cloud. (6) Overlapping map by adding detected groove region and ground truth of groove region. Eventually the overlapping map is used to compute 3D overlapping rate to evaluate accuracy of the groove detection algorithm.

C. Motion Execution Results

As discussed in Section IV, the motion trajectory generation is based on the point set of the detected groove region as shown in Fig. 16. Therefore the robotic manipulator drives the welding torch following the motion trajectory to execute welding tasks. Fig 17 presents actual motion execution of the robotic manipulator.

As shown in Fig. 17, the motion of the manipulator fits the generated trajectory tightly without terrible mismatch. And due to cartesian path planning of the Moveit [23] package,

TABLE II

RESULTS OF GROOVE DETECTION ACCURACY FOR EACH WORKPIECE

Types	λ_1	λ_2	λ_3	λ_4	λ_5	λ_{mean}
Straight-line	92.74	93.01	92.81	92.49	91.35	92.48
Curve-line	81.24	82.58	82.13	81.87	82.29	82.02
Box	81.48	82.65	82.02	80.49	81.97	81.72
Cylinder	63.27	61.79	64.69	67.57	65.75	64.61

λ_1-λ_5 = Accuracy (%) of each test. λ_{mean} = Average accuracy.

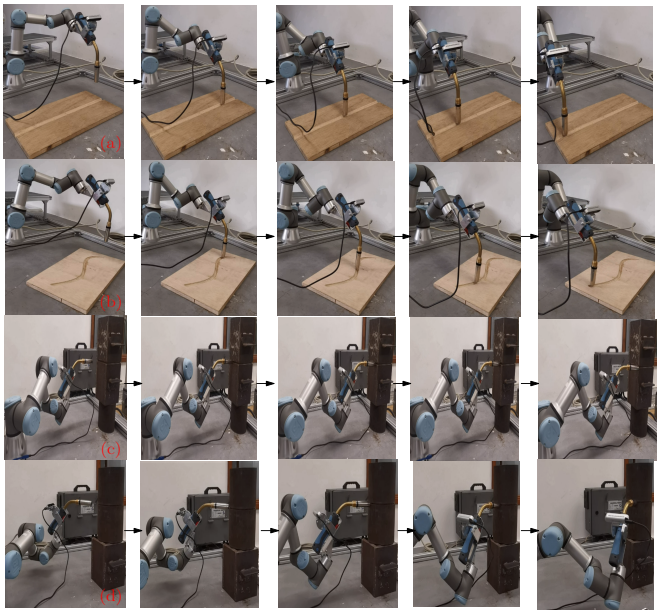

Fig. 17. Actual motion execution of the robotic manipulator. (a)-(b) represents type of straight-line, curve-line, metal box and metal cylinder workpiece respectively. And their motion order is from left side to right side.

the welding torch is able to complete welding tasks well (smoothly following the trajectory of groove).

According to running performance (runtime and detection accuracy) of the proposed method, the robot system has capability of realizing automatic welding tasks compared with traditional teach and playback method. Although the groove detection and trajectory planning method takes up most of computational resources, it is acceptable for the actual welding.

VI. CONCLUSIONS

This paper presents an integrated robot system for industrial welding with an automatic groove detection and trajectory planning method. The system is composed of a robotic manipulator (Universal Robot 3), an RGB-D camera (Realsense D415) and a welding torch. And the system has good flexibility of facing different welding situation. The software framework is totally built up on ROS and 3D point cloud processing is the key part. In real experiment four types of general welding workpiece are tested. Through evaluating accuracy between welding trajectory generated by the proposed method and the ground truth of welding groove, the motion execution performance proves good feasibility of

the designed robot system. The current problem is that the proposed method can not cope with larger 3D point cloud of welding workpiece surface due to more complex geometrical region and noise. In future work, neural network instead of geometrical feature based method may be introduced to improve robustness and accuracy of welding groove detection algorithm. And more important work is that the improved robot system would be implemented into actual industrial welding tasks.

REFERENCES

[1] Y. Li, Y. F. Li, Q. L. Wang, D. Xu, and M. Tan, "Measurement and defect detection of the weld bead based on online vision inspection," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 7, pp. 1841–1849, 2009.

[2] Y. Liu and Y. Zhang, "Iterative local anfis-based human welder intelligence modeling and control in pipe gtaw process: A data-driven approach," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 3, pp. 1079–1088, 2014.

[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 Transactions on Industrial Electronics*, vol. 64, no. 9, pp. 7201–7209, 2017.

[4] J. Li, F. Jing, and E. Li, "A new teaching system for arc welding robots with auxiliary path point generation module," in *2016 35th Chinese Control Conference (CCC)*. IEEE, 2016, pp. 6217–6221.

[5] M. Rodríguez-Martín, P. Rodríguez-Gonzálvez, D. Gonzalez-Aguilera, and J. Fernandez-Hernandez, "Feasibility study of a structured light system applied to welding inspection based on articulated coordinate measure machine data," *IEEE Sensors Journal*, vol. 17, no. 13, pp. 4217–4224, 2017.

[6] S. M. Ahmed, Y. Z. Tan, G. H. Lee, C. M. Chew, and C. K. Pang, "Object detection and motion planning for automated welding of tubular joints," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 2610–2615.

[7] H. Zhou, H. Cui, and Q. H. Qin, "Influence of ultrasonic vibration on the plasticity of metals during compression process," *Journal of Materials Processing Technology*, vol. 251, pp. 146–159, 2018.

[8] N. Lv, Y. Xu, S. Li, X. Yu, and S. Chen, "Automated control of welding penetration based on audio sensing technology," *Journal of Materials Processing Technology*, vol. 250, pp. 81–98, 2017.

[9] J. Zhu, J. Wang, N. Su, G. Xu, and M. Yang, "An infrared visual sensing detection approach for swing arc narrow gap weld deviation," *Journal of Materials Processing Technology*, vol. 243, pp. 258–268, 2017.

[10] W. J. Shao, Y. Huang, and Y. Zhang, "A novel weld seam detection method for space weld seam of narrow butt joint in laser welding," *Optics & Laser Technology*, vol. 99, pp. 39–51, 2018.

[11] H. Ma, S. Wei, Z. Sheng, T. Lin, and S. Chen, "Robot welding seam tracking method based on passive vision for thin plate closed-gap butt welding," *The International Journal of Advanced Manufacturing Technology*, vol. 48, no. 9-12, pp. 945–953, 2010.

[12] S. H. Rao, V. Kalaichelvi, and R. Karthikeyan, "Real-time tracing of a weld line using artificial neural networks," in *2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*. IEEE, 2018, pp. 275–280.

[13] L. Nele, E. Sarno, and A. Keshari, "An image acquisition system for real-time seam tracking," *The International Journal of Advanced Manufacturing Technology*, vol. 69, no. 9-12, pp. 2099–2110, 2013.

[14] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 1–4.

[15] Y. Xu, N. Lv, G. Fang, S. Du, W. Zhao, Z. Ye, and S. Chen, "Welding seam tracking in robotic gas metal arc welding," *Journal of Materials Processing Technology*, vol. 248, pp. 18–30, 2017.

[16] S. M. Ahmed, Y. Z. Tan, C. M. Chew, A. Al Mamun, and F. S. Wong, "Edge and corner detection for unorganized 3d point clouds with application to robotic welding," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 7350–7355.

[17] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3d object detection from point clouds," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7652–7660.

[18] S. Song and J. Xiao, "Sliding shapes for 3d object detection in depth images," in *European conference on computer vision*. Springer, 2014, pp. 634–651.

[19] P. Maiolino, R. Woolley, D. Branson, P. Benardos, A. Popov, and S. Ratchev, "Flexible robot sealant dispensing cell using rgb-d sensor and off-line programming," *Robotics and Computer-Integrated Manufacturing*, vol. 48, pp. 188–195, 2017.

[20] L. Jing, J. Fengshui, and L. En, "Rgb-d sensor-based auto path generation method for arc welding robot," in *2016 Chinese control and decision conference (CCDC)*. IEEE, 2016, pp. 4390–4395.

[21] V. Patil, I. Patil, V. Kalaichelvi, and R. Karthikeyan, "Extraction of weld seam in 3d point clouds for real time welding using 5 dof robotic arm," in *2019 5th International Conference on Control, Automation and Robotics (ICCAR)*. IEEE, 2019, pp. 727–733.

[22] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[23] S. Chitta, I. Sucan, and S. Cousins, "Moveit![ros topics]," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 18–19, 2012.

[24] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3d recognition and pose using the viewpoint feature histogram," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 2155–2162.