Extraction of Weld Seam in 3D Point Clouds for Real Time Welding Using 5 DOF Robotic Arm

Vrushali Patil[*] , Indraneel Patil[†], V.Kalaichelvi[‡], R. Karthikeyan[§]
[*] [†‡]Department of Electrical and Electronics Engineering
[§]Department of Mechanical Engineering
BITS Pilani, Dubai Campus, Dubai, UAE
e-mail: f20160079@dubai.bits-pilani.ac.in[*] , f20150141@dubai.bits-pilani.ac.in[†]
kalaichelvi@dubai.bits-pilani.ac.in[‡], rkarthikeyan@dubai.bits-pilani.ac.in[§]

Abstract—**Welding involves the union of two workpiece halves along a common edge of interface, also called as the weld seam. In this paper a novel algorithm has been proposed which is independent of the shape of the workpiece to classify and extract weld seams from 3D point clouds. The real time tracking of the weld seam is also demonstrated on a 5 axis robotic manipulator. The aim was to eliminate the need for manual computation of the robot trajectory and also ensure that the overall pipeline is computationally efficient. The two workpiece halves have been clustered into different point clouds and the edge of interface on either of the halves has been specified using the centroid shift method. The precision recall metrics of our algorithm were very close to the ideal values on any shape of the weld seam. Singularity avoidance, trajectory planning, kinematics of the robotic manipulator is beyond the scope of this paper.**

Keywords-robotic manipulator; point cloud; weld seam; artificial neural networks

I. INTRODUCTION

Assuming the various advantages of robotic welding which have been introduced in brief in the related works section, in this paper the primary focus is on 3D weld seams applicable to any common robotic machining task. One of the main problems here is extracting the weld seam autonomously and accurately irrespective of the shape of the welding workpiece. It was observed that all previous solutions required manual input from the user to highlight the weld seam which is then later sent to the manipulator as a 3D trajectory. Our work eliminates the need for any human interference in this problem and enables a real time solution to extract and track a weld seam by a 5 DOF manipulator. We have mainly used the Point Cloud Library to implement our algorithm, ROS robotics framework for data acquisition and communication and the Matlab Neural Networks Toolbox for transformation of coordinate axes.

The objectives and contributions of this paper are:

- To propose a PCL pipeline which automatically extracts the weld seam from a 3D point cloud so that the manipulator trajectory does not have to be manually specified using a CAD model.
- Proposed methodology can be used for finding 3D pose of weld seam, seam tracking and measuring dimensions of the weld seam.
- To ensure that no restrictions are imposed on the 3D shape of the weld seam or the geometrical shape of the welding workpiece.
- To ensure real time tracking so that pipeline also works for a dynamically moving work piece.
- To test the pipeline and compile experimental results using a 5 DOF industrial robotic manipulator and a Kinect stereo camera.

The paper is divided into six sections. The second section discusses the novel proposed PCL methodology for automatic extraction of the weld seam. The third section consists of the description of the hardware experimental set up. The fourth section consists of tracking the extracted weld seam using a robotic manipulator in real time. The fifth section compiles all our results, evaluates the performance of both the algorithm and the motion planning and validates our claims. The final section summarizes the findings, contributions and the future scope of this work.

II. RELATED WORK

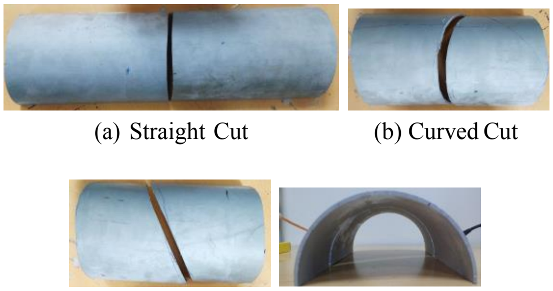
With the ongoing advancement in optimal control, trajectory planning and modern ways of sensor data processing robotic manipulators have become extremely mainstream in all types of industry machining applications like robotic welding. There are many advantages of using robotic machining cells over other conventional machines like the Computer Numerical Control machine especially while considering the much wider working area which allows it to accommodate much bigger workpieces and the flexible kinematics of robotic arms, are often capable of machining parts with intricate details and complex shapes [1]. Despite these obvious wins for robotic machining some common criticism about their performance and efficiency state the following facts: Low positioning accuracy during complex tasks resulting in far from robust operation More prone to disturbances from reaction forces on the end-effector tool Possibility of resonance due to machining process vibrations

978-1-7281-3326-3/19/$31.00 ©2019 IEEE

Hence for robotic machining cells to take over structured tasks, it is a given that real time closed loop control and advanced sensors are required to overcome the above shortcomings. 2D computer vision is proven as an extremely useful tool to assist robotic machining tasks like welding in the following literature 2, 3, 4 and 5. However, in the recent times researchers have commonly preferred stereo cameras over normal cameras while implementing such complex tasks, due to the advancement in research in 3D computer vision using the Point Cloud Library (PCL) [6], emergence of low cost devices like the Kinect and since point clouds capture more data than images including the depth perception. Use of Stereo cameras in an ordered task like welding also eliminates the need for a fixed welding work-table, allows to place no restriction on the shape of the workpiece or the weld line and gives freedom to use the entire workspace of robotic manipulator [7]. Since the aim of the paper is to automatically segment the weld seam using 3D computer vision, the subject of edge segmentation in complex point clouds gains special importance. It usually serves as the foundation for solving higher level problems such as object recognition, interaction analysis and scene understanding [8]. Edge based segmentation is quite compli- cated since there are many different types of edges in any point cloud just like any 2D image. Depth discontinuities in the 3D data produce two related types of edges: occluding and occluded, areas where surface normals change rapidly lead to high curvature edges, RGB edges are obtained by backprojecting edges on the 2D photometric data onto the 3D structure[9]. The most revolutionary papers in this area are edge detection by analyzing the geometric properties of each query point's neighborhood, and then combining Random Sample Consensous (RANSAC) and angular gap metric [10], edge feature detection by computing a Gauss map clustering on local neighborhoods and then an iterative feature selection process [11], contour extraction by setting a contour score for each individual point as predicted with a binary classifier using a set of features extracted from the point's neighborhood and then selecting an optimal set of contours from the candidates [12] and more recently edge detection using the fast nearest neighbor search on organized point clouds [13]. Most of the above methods are either restricted to organized point clouds or are unsuitable for real time application due to the high time delay. Hence real time operation of the robotic manipulator is not possible. For Unorganized point clouds most recent and according to us the only technique feasible for real time robotic application and also includes online trajectory generation is based on classifying points as edge or non-edge based on evaluating degree of shift in centroid from their initial position [14]. But an important aspect which has not been addressed in this paper is how the weld seam is extracted from all the boundary edges of the welding workpiece. In the experimental results sections, the authors mention that the panel contains only straight line seams but in all practical applications, the weld seam must be extracted from all different types of edges in a variety of workpiece shapes. We have successfully taken up this challenge and found a solution using a novel approach in our paper.

III. Methodology

Organized point cloud of the workspace is the starting point and the end goal of this PCL pipeline is to extract a weld line from the noisy point cloud which can be later sent to the robotic manipulator as a 3D trajectory for tracing the weld seam. It is assumed that the welding workpiece is kept inside the workspace of the manipulator. Pipeline is described here with a cylindrical welding workpiece with a radially cut welding seam which is shown in Fig. 1. The entire pipeline can be divided into two branches. Due to limitations in computation power, the input cloud has been downsampled to unorganized point cloud in Branch 1. Branch 2 describes the rest of the organized point cloud processing to extract the weld seam. The algorithm flowchart is shown in Fig. 2.


(a) Straight Cut

(b) Curved Cut

(c) Inclined Cut

(d) Cross-section

Figure 1. Workpieces used for testing.

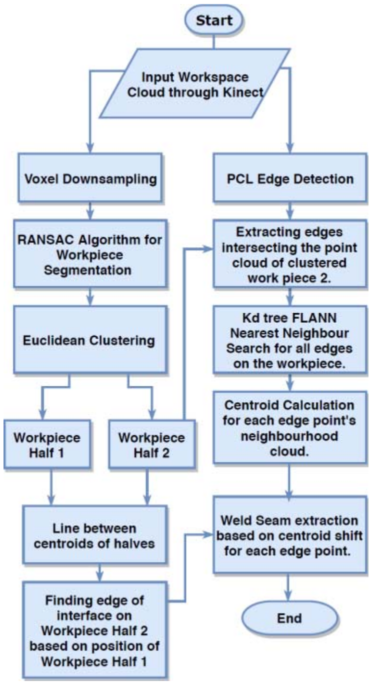
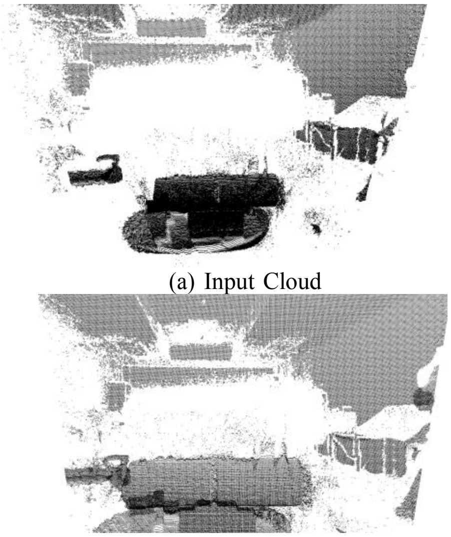

Start
Input Workspace Cloud through Kinect
Voxel Downsampling
RANSAC Algorithm for Workpiece Segmentation
Euclidean Clustering
Workpiece Half 1
Workpiece Half 2
Line between centroids of halves
Finding edge of interface on Workpiece Half 2 based on position of Workpiece Half 1
PCL Edge Detection
Extracting edges intersecting the point cloud of clustered work piece 2.
Kd tree FLANN Nearest Neighbour Search for all edges on the workpiece.
Centroid Calculation for each edge point's neighbourhood cloud.
Weld Seam extraction based on centroid shift for each edge point.
End

Figure 2. Algorithm flowchart.

A. Branch 1

1) Voxel Downsampling: Aim of this step is to reduce the number of points in the point cloud so that the processing is faster and pseudo real time. PCL VoxelGrid class creates a 3D voxel grid in the form of boxes over the input point cloud data and then in each box all the points present are replaced by their centroid. Unluckily organized point cloud dataset is the one in which data is organized into a fixed number of rows and columns and hence once the cloud has been downsampled, it is converted into an unorganized cloud making any type of computation all the more challenging. The input cloud and the downsampled cloud are shown in Fig. 3(a) and Fig. 3(b) respectively.


(a) Input Cloud

(b) Downsampled Cloud

Figure 3. Prefiltering of cloud.

2) Random Sample Consensus (RANSAC) Workpiece Segmentation: In this step the welding workpiece was segmented from the entire workspace of the manipulator so that the noisy points, other objects and obstacles can be rejected from the input point cloud (Fig. 4). The RANSAC model iterates through different Sample Consensus Models namely spherical, planar and cylindrical, tries to estimate model of work piece and then segments out the workpiece from the cloud [15].

RANSAC algorithm for segmenting a geometrical object in point clouds can be summarized with the following steps:

- A random subset of points from the entire point cloud are considered as the hypothetical inliers for the geometrical model.
- All the coefficients of the geometric model are estimated using these inlier points in the cloud.
- Then all the other points are iterated and added to those points which fit this model to the inliers list.
- The model is re-estimated from the final list of inliers once some satisfactory threshold of inlier points has been reached [16].

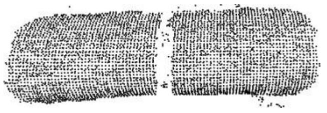


Figure 4. Segmented workpiece.

3) Euclidean Clustering: After segmenting the workpiece, the two halves that are to be welded together are clustered individually and pushed to two separate point clouds. In this step weld seam formed from the interface edge of two halves is converted into two separate occluding edges. The focus is now on the individual occluding edges in either of the workpiece halves. Euclidean Clustering is utilized to separate the two work piece halves into two separate clusters. The goal of the clustering technique is to divide unorganized point cloud model into smaller parts based on spatial decomposition. An object cluster is defined as follows, where dth is maximum imposed distance threshold: Let $O_i = \{p_i \in P\}$ be a distinct cluster from $O_j = \{p_j \in P\}$ if:

$\min \|p_i - p_j\|^2 \geq d_{th}$ ... (1)

All the steps in Euclidean Clustering can be summarized as follows:

- A Kd-tree representation is superimposed on input point cloud P, C is the empty list of clusters and a Q is the array of points that need to be clustered.
- Perform the following steps as we cycle through each point $p_i \in P$:
  - Add p_i to the array of points Q and for every point $p_i \in Q$ do:
  - Search for the set P^i_k of point neighbors of p_i in a sphere with radius $r < d_{th}$;
    - For every neighboring point $p^k_i \in P_i^k$, which is unprocessed, add it to Q;
- Once every point in Q is classified, add Q to the list of clusters C, and reset Q
- Euclidean clustering ends once all points $p_i \in P$ are clustered into the clusters list C

4) Identifying the Interface Edge: After Euclidean Clustering, we accessed two different workpiece halves and saved them in two different point clouds. Further, only one cluster of the workpiece which is half 2 (right) was considered. In this right half, we must reject all the other edges not related to the edge of interface which forms the weld seam along with its other half. For achieving this, it is required to find how the left cluster (half 1) is oriented and translated with respect to this particular half 2. Assuming the current workpiece half 2 has four edges: top, bottom, left and right. Any one of these four edges could be the interface edge along which the weld seam is present. To identify the pose of the second cluster, the PCL Centroid Class was used to independently find centroids of both the workpiece halves. Furthermore, both the spatial centroid points were joined with a 3D line (Fig. 5). Based on the relative location of both the centroids and by comparing x and y coordinates it was concluded which of the edges is the edge of interface. Rest of the edges are not required and can be rejected. In this case the left edge of the workpiece half 2 is the edge of interface and the top, bottom and right edges can be ignored.

B. Branch 2

1) Scene Edge Segmentation : The most reliable and real time algorithms for edge segmentation in PCL require or- ganized point clouds. So all the Branch 2 operations were performed on the default input point cloud of the workspace scene. Reliable 3D edges are computed by exploiting depth discontinuities, photometric RGB intensity changes and by backprojecting 2D edges to 3D point clouds instead of complex 3D processing [9]. Out of all the types of edges, high curvature edges were chosen to give the best results. High curvature edges occur where surface normals change abruptly. Apply a variant of Canny Edge Detection, the regions with higher degree of normal variations are detected by applying Sobel operator to the x and y components of the normals which are detected from the point clouds. At the end of this step we have a point cloud of high curvature edges of the entire scene (Fig. 6).

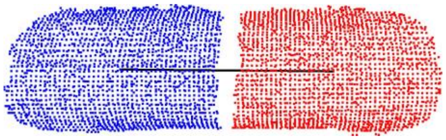


Figure 5. Line joining the two workpiece centroids.

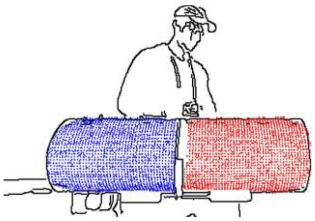
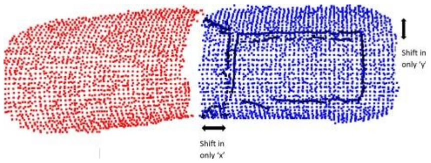


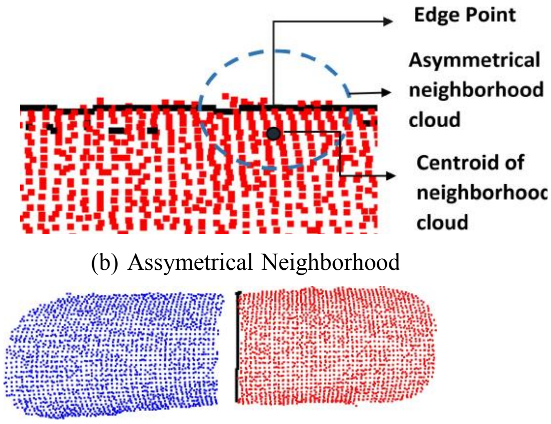
Figure 6. Edges in the input cloud.

2) Workpiece Edge Segmentation: As it was required in finding the edges for the welding workpiece, workpiece edges from the entire scene edges were isolated. As discussed earlier, workpiece half 2 from Step 3 in Branch 1 was considered as the target cloud. All the rest of our processing was done only on this half. It is possible to isolate the edges related to the workpiece half 2 through the PCL extract indices class directly. But since the workpiece half cluster from Branch 1 is an unorganized point cloud and the high curvature edge cloud is an organized one, hence it was required to use alternative methods to find the edges. The KdtreeFLANN PCL class was used for performing fast approximate nearest neighbor searches in the high curvature edges cloud around each point in the workpiece half cloud. Kd-tree is a data structure in computer science and is a binary search tree where constraints are imposed on it to perform nearest neighbor searches. All the points in the high curvature cloud which are declared inliers to our search radius of 0.1 were exported to another cloud called workpiece half edges.

3) Edge of Interface Extraction based on Centroid Shift: The workpiece half edges cloud was available and also it was known from Branch 1 that the left edge of the workpiece half 2 is the interface edge. Since in point clouds there are no individual edges but just a grid of points, we were required to extract all the points corresponding to this left edge from the workpiece half 2 edges cloud. The left edge points from the workpiece_half_edges cloud was computed using the Centroid Shift Approach. The KdTreeFLANN nearest neighbor search PCL class was employed again to find the neighbors of each point in the workpiece_half_edges cloud. These nearest neighbors were exported to temporary cloud temp. Since cloud temp is a cloud consisting of neighbors of an edge point, it is asymmetrically distributed. If the centroid of this temp cloud is calculated, it shifts away from the edge towards the center of workpiece half 2. This shown in Fig. 7(a) and Fig. 7(b). This is the fact that has been exploited to classify the edge point as top, bottom, left or right edge point. Since the edge of interest lies to the left side of the workpiece half 2, it can be easily concluded that the centroid of the neighbors cloud of every left edge point will shift towards the right. We iterated through every point in the workpiece_half_edges cloud. Further, the difference between the co-ordinates of the edge point and the centroid point of the neighborhood cloud temp was computed. For a top edge, only the y coordinate of the centroid varied while the x and z co-ordinates remained almost constant. Similarly, for a left edge, the difference between the y and z co-ordinates was zero while the x co- ordinate had shifted significantly by some distance. Thus, the points corresponding to the edge of interface were extracted and pushed to the final segmented weld_seam cloud. This left edge of the workpiece half 2, which is in principle an occluding edge is the interface edge between the two halves of the welding workpiece that needs to be welded together. This interface edge also accurately overlaps our desired weld seam in which we are interested in. Thus, the goal of extracting the weld seam in real time in a computationally efficient way was achieved (Fig. 7(c)).



(a) Shifted Centroids Along Edge


Edge Point
Asymmetrical neighborhood cloud
Centroid of neighborhooc cloud

(b) Assymetrical Neighborhood

(c) Segmented Weld Seam

Figure 7. Segmentation of weld seam from all the edges.

IV. Hardware Experimental Setup

The 5-DOF TAL Brabo! manipulator was employed to trace the weld seam extracted by our proposed pipeline. This manipulator does not allow to communicate with the servo motors directly rather trio motion controller is used to control the motors of the manipulator by preset factory conditions. Furthermore, Trio motion controller provides only ActiveX support to interface the robot to the controller. The ActiveX control standard is developed by Microsoft and it describes modular, reusable software components that can be used universally by any environment that supports the standard and in any programming language. MATLAB was identified as the best way to interact with the controller through this ActiveX COM object. An ActiveX window was created and connection was established using the function 'actxcontrol' supported by MATLAB so that the robotic arm can be controlled using different public properties and methods of the PC Motion COM object. The entire hardware setup is shown in Fig. 8.

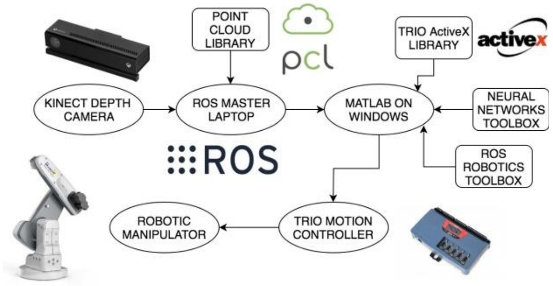


Figure 8. Experimental setup.

To create a real time welding process, it is necessary to communicate continuously with the robotic manipulator. The information about the control actions and movement commands should be communicated to the controller of the manipulator with high speed and accuracy. Hence, the widely used Robot Operating System (ROS) framework was adopted for data acquisition and data communication [18]. ROS is a meta-operating system which runs on top of a Linux computer and helps in the intercommunication of different input-output devices with minimal lines of code. A Microsoft Kinect Sensor was placed in workspace and data from it was accessed using the iai_kinect2 ROS package [19]. The entire point cloud processing and proposed pipeline was implemented on a Linux computer with DDR4 SDRAM 2133Mhz 16GB, 512GB SSD, 6th Gen Intel Core i7 and in C++ language. However, it was also required to have a windows operating system for the Microsoft ActiveX library, hence one Linux computer acquired data from the sensor and processed it while the windows computer controlled the robotic arm. ROS cannot be installed on Windows computers. Hence in the present work, ROS in MATLAB was employed to allow MATLAB/ Simulink to communicate with other ROS computers on the same network. Further, both the Linux computer and the Windows computer were connected using ethernet cables and connection was enabled so that data can be seamlessly transferred between each other. Each computer was made available and visible in the ROS network for bilateral connectivity. The Linux computer published data while the windows computer subscribed this data and performed further actions. After successful extraction of the edge, x, y, z co-ordinates of the weld seam were sent from the Linux computer to the windows computer running MATLAB through ROS multiple machines setup. This data was then fed to the neural network and real world co-ordinates were obtained. The output co-ordinates of the network were sent to the controller and the robotic arm traced the weld seam in real time.

V. Robotic Manipulator Motion Planning

As mentioned earlier, the TAL Brabo! robotic arm does not allow implementation of inverse kinematics to control its motion. Therefore, Artificial Neural Networks (ANN) were employed to translate the point cloud x, y, z co-ordinates to the real world X, Y, Z co-ordinates, which can be sent to the trio controller [5]. An artificial neuron is a computational model which consists of nonlinear statistical data modeling tools where the complex relationships between inputs and outputs are modeled or patterns are found. The fundamental processing element of a neural network is a neuron. The weights of these neurons are adjusted on the basis of a given data. In first feedforward pass of the neural network training, random weights and biases are assigned to the neurons. Later, at every epoch of the backpropagation pass, these weights are updated so that the error metric is minimized.

A. Data Collection and Architecture

Supervised learning for ANN's was employed where input and output vectors were rendered to the network. A hexagonal plane was segmented by the RANSAC algorithm. Further, centroid of the segmented plane was computed and pushed to the point cloud. The x, y and z co-ordinates of the centroid were recorded from the point cloud of the plane. These centroid co-ordinates served as input to the Neural Network. The robotic arm was then manually moved to the center of the plane and real world coordinates were recorded as target data. Furthermore, the plane was moved and held at different locations, heights and orientations and 50 readings were recorded so that the workspace of the robotic arm was covered majorly. The position of the Kinect sensor was fixed to avoid errors. The neural network consisted of 3 neurons in the input layer, 9 neurons in the hidden layer and 3 neurons in the output layer. The architecture is shown in figure below. After processing the input point cloud, the weld seam was extracted by the proposed pipeline. The x, y, z co-ordinates of this segmented weld line were published by the Linux computer. The windows computer received these co-ordinates in MATLAB through ROS Subscriber in real time. These end- coordinates of the weld seam were input values to the Neural Network. The pre-trained Neural Network model then mapped the point cloud co-ordinates to the X, Y, Z co-ordinates of the end effector. The tracing of the weld seam was experimentally verified by sending the output values (X, Y, Z) from the network to the trio motion controller of the robotic arm. The robotic arm moved to these co-ordinates and traced the weld seam.

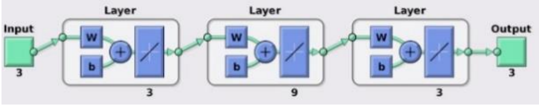


Figure 9. Neural network architecture.

VI. Experimental Results and Analysis

A. Evaluation Metrics of the Proposed Weld Seam Extraction

The proposed weld seam extraction pipeline was evaluated on three different welding scenarios. In all three scenarios, the welding workpiece was always cylindrical in shape but the welding seams were cut radially straight, radially inclined and radially curved. The pipeline is independent of the shape of the workpiece and will work flawlessly on any geometrical shape spherical, cylindrical, conical or planar. The obtained results are shown in the Fig. 10 below.

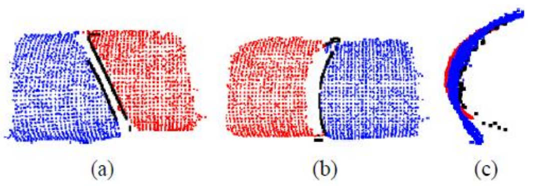


Figure 10. (a) Inclined Cut (b) Curved Cut (c) Cross-section.

We have chosen precision and recall estimates as the evaluation metric in analyzing the performance of our pipeline in all the three scenarios. Ground truth of the weld seam has been calculated from the 3D point cloud by manually selecting the points along the desired trajectory in Mesh Lab [20].

$$Precision = \frac{T\ \ P}{TP + FP} \qquad Recall = \frac{T\ \ P}{TP + FN}$$

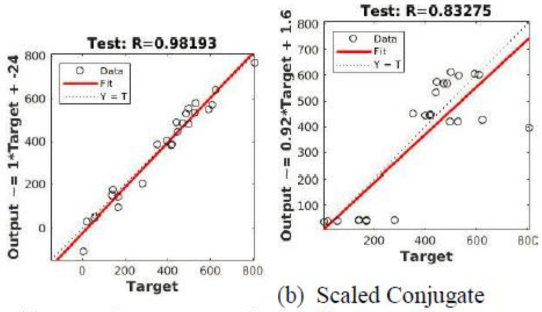
where TP stands for True Positives which are correctly identified points, FP stands for False Positives for wrongly detected points, FN stands for False Negatives, representing false rejections, i.e. points that belong to the weld seam in the ground truth but are not extracted by our PCL pipeline [21]. In the present work we have obtained the optimal performance of the proposed methodology with Cluster Tolerance of 0.8 cm, 0.1 cm radius of sphere for nearest neighbor search while calculating shift in centroid and 0.01 cm as threshold for centroid shift.

TABLE I. Precision and Recall Metrics

	True Positives	False Positives	False Negatives	Precision	Recall
Radially Straight	107	3	11	0.97272	0.90677
Radially Inclined	102	24	12	0.809523	0.89473
Radially Curved	87	26	26	0.769911	0.76991

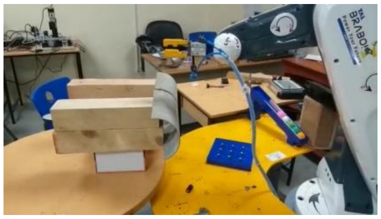
B. Performance of the Motion Planning Algorithm

Levenberg Marquardt (LM), and Scaled Conjugate Gradient (SCG) algorithms were used to train and compare the performance of the neural networks keeping all the parameters constant for comparison. The data was divided into following ratio: Training set: 65LM produced least mean squared error and fit more data points as compared to the SCG algorithm (Figure 11). This algorithm also performed well on unseen data. The error on unseen data was very less, hence Levenberg Marquardt (LM) with a learning rate of 0.01 was chosen for the neural network training. To verify the performance of the algorithm in real time, we conducted several experiments. The path traced by the robotic arm was satisfactory and it can be concluded that the motion of the arm can be controlled using Neural Networks.

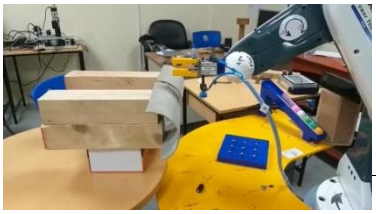


(a) Levenberg Marquardt (b) Scaled Conjugate Gradient

Figure 11. Performance of different algorithms.



(a)



(b)

Figure 12. Real time tracing using 5 DOF robotic arm.

VII. Conclusion and Future Work

In this paper, a novel approach has been demonstrated to extract the 3D weld seam from noisy background. Experiments were performed with different weld seam curves. The precision obtained was very close to ideal readings. It can be concluded that the proposed pipeline works for all types of weld cuts and also allows placement of the workpiece in any orientation and alignment so as to utilize the entire 3D workspace of the robotic arm. Further, using MATLAB and Neural Networks, the control of robotic arm to trace the weld seam is presented. The performance of the neural network was analyzed by tuning various parameters. Levenberg Marquardt algorithm was chosen as it performed very well on unseen data. The robot traced the weld seam in real time for all three weld seam profiles with satisfactory results. However, one limitations of our algorithm is the requirement of organized point cloud. In present work, two separate branches are considered for processing and are later combined. This work can be extended such that the entire processing of weld seam is carried out on unorganized point cloud. Secondly, the depth sensor is fixed at a static position in the workspace of the robotic arm. The dynamic welding process can be implemented by placing the Kinect sensor on the robotic arm. In this paper forward kinematics is employed. An attempt can be made to circumvent the given controller of the robotic arm so that inverse kinematics can be adopted.

Acknowledgment

The authors are extremely grateful for the constant support, encouragement and motivation given to us by the authorities, colleagues and friends at BITS Pilani, Dubai Campus. We are also thankful for all the hardware and software facilities provided to us for carrying out this research work.

References

[1] Doukas, Christos & Pandremenos, Ioannis & Stavropoulos, Panagiotis. (2011). Machining with Robots: A Critical Review.

[2] Liu Suyi, Liu Lingteng, Zhang Hua, Bai Jianjun and Wang Guorong, ”Study of robot seam tracking system with laser vision,” 2009 Interna- tional Conference on Mechatronics and Automation, Changchun, 2009, pp. 1296-1301.

[3] Ma, Hongbo & Wei, Shanchun & Sheng, Zhongxi & Lin, Tao & Chen, Shanben. (2010). Robot welding seam tracking method based on passive vision for thin plate closed-gap butt welding. The International Journal of Advanced Manufacturing Technology. 48. 945-953. 10.1007/s00170- 009-2349-4.

[4] L. A. Fernandes, A. Saraiya, V. Kalaichelvi and R. Karthikeyan, ”Performance Evaluation of Digital Image Processing and Artificial Neural Networks for Weld Line Detection of Robotic Manipulator,” 2017 9th IEEE-GCC Conference and Exhibition (GCCCE), Manama, 2017, pp. 1-6.

[5] S. H. Rao, V. Kalaichelvi and R. Karthikeyan, ”Real-Time Tracing Of A Weld Line Using Artificial Neural Networks,” 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS), Singapore, 2018, pp. 275-280.

[6] R. B. Rusu and S. Cousins, ”3D is here: Point Cloud Library (PCL),” 2011 IEEE International Conference on Robotics and Automation, Shanghai, 2011, pp. 1-4.

[7] 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,” 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, 2016, pp. 2610-2615.

[8] X. Lin, J. R. Casas and M. Pardas, ”3D point cloud segmentation using a fully connected conditional random field,” 2017 25th European Signal Processing Conference (EUSIPCO), Kos, 2017, pp. 66-70.

[9] C. Choi, A. J. B. Trevor and H. I. Christensen, ”RGB-D edge detection and edge-based registration,” 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, 2013, pp. 1568-1575.

[10] Ni, Huan & Lin, Xiangguo & Ning, X & Zhang, Jixian. (2016). Edge Detection and Feature Line Tracing in 3D-Point Clouds by Analyzing Geometric Properties of Neighborhoods. Remote Sensing. 8. 710. 10.3390/rs8090710.

[11] C. Weber, S. Hahmann and H. Hagen, ”Sharp feature detection in point clouds,” 2010 Shape Modeling International Conference, Aix-en-Provence, 2010, pp. 175-186.

[12] T. Hackel, J. D. Wegner and K. Schindler, ”Contour Detection in Unstructured 3D Point Clouds,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 1610- 1618.

[13] R. Bormann, J. Hampp, M. H a¨gele and M. Vincze, ”Fast and accurate normal estimation by efficient 3d edge detection,” 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, 2015, pp. 3930-3937.

[14] Ahmed, Syeda Mariam, Yan Zhi Tan, Chee-Meng Chew, Abdullah Al Mamun and Fook Seng Wong. “Edge and Corner Detection for Unorganized 3D Point Clouds with Application to Robotic Welding.” CoRR abs/1809.10468 (2018): n. pag.

[15] Schnabel, Ruwen & Wahl, Roland & Klein, Reinhard. (2007). Efficient RANSAC for point-cloud shape detection. Comput. Graph. Forum. 26. 214-226. 10.1111/j.1467-8659.2007.01016.x.

[16] A. Fischler, Martin & C. Bolles, Robert. (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications To Image Analysis and Automated Cartography. Communications of the ACM. 24. 381-395. 10.1145/358669.358692.

[17] C. Choi, A. J. B. Trevor and H. I. Christensen, ”RGB-D edge detection and edge-based registration,” 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, 2013, pp. 1568-1575.

[18] Quigley, M & P. Gerkey, B & Conley, K & Faust, J & Foote, T & Leibs, J & Berger, E & Wheeler, R & Y. Ng, A. (2009). ROS: An open-source Robot Operating System. ICRA Workshop on Open Source Software. 3. 1-6.

[19] T. Wiedemeyer, “IAI Kinect2,” https://github.com/code-iai/iai kinect2, Institute for Artificial Intelligence, University Bremen, 2014 – 2015, accessed June 12, 2015.

[20] Cignoni, Paolo & Callieri, Marco & Corsini, Massimiliano & Dellepiane, Matteo & Ganovelli, Fabio & Ranzuglia, Guido. (2008). Mesh- Lab: an Open-Source Mesh Processing Tool.. Computing. 1. 129-136. 10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136.

[21] Raghavan, Vijay & S. Jung, Gwang & Bollmann, Peter. (1989). A Critical Investigation of Recall and Precision as Measures of Retrieval System Performance.. ACM Transactions on Information Systems. 7. 205-229. 10.1145/65943.65945.