

# CNN-based Feature Extraction for Robotic Laser Scanning of Weld Grooves in Tubular T-joints

Øyvind W. Mjølhus<sup>1</sup>, Andrej Cibicik<sup>2</sup>, Eirik B. Njaastad<sup>2</sup> and Olav Egeland<sup>1</sup>

**Abstract**—This paper presents an algorithm for feature point extraction from scanning data of large tubular T-joints (a sub-type of a TKY joint). Extracting such feature points is a vital step for robot path generation in robotic welding. Therefore, fast and reliable feature point extraction is necessary for developing adaptive robotic welding solutions. The algorithm is based on a Convolutional Neural Network (CNN) for detecting feature points in a scanned weld groove, where the scans are done using a laser profile scanner. To facilitate fast and efficient training, we propose a methodology for generating synthetic training data in the computer graphics software Blender using realistic physical properties of objects. Further, an iterative feature point correction procedure is implemented to improve initial feature point results. The algorithm's performance was validated using a real-world dataset acquired from a large tubular T-joint.

**Index Terms**—laser scanner, convolutional neural network, synthetic data, weld groove.

## I. INTRODUCTION

Robotic welding is often considered an efficient and safe alternative to manual welding. A typical industrial application is the mass production of relatively precise mechanical parts, where the robot is pre-programmed offline for a repetitive welding task. In this case, robots cannot adapt to variations in the geometry of a workpiece, an issue particularly prevalent when working with large structures. A particular case is robotic welding of large tubular T-joints (see Fig. 1), where multi-layer welds are done in deep machined weld grooves. Tubular T-joints (and TKY-joints, in general) constitute many hours of manual welding work in constructing large offshore structures. Robotic welding can, for such cases, benefit from the geometric information of a groove being collected by vision systems, such as laser triangulation sensors.

A common type of structured light sensor used for robotic welding is the laser line scanner, where the projected light is a single straight line [1], [2], and the output is a set of triangulated points. The subsequent process of extracting a small subset of points from the measurements sufficiently describing the geometry of the groove is generally referred to as feature extraction. The nature of the feature points may vary based on the category of weld considered; butt joints, lap joints or stub joints adhere to different geometric patterns, and



Fig. 1. An example of a large tubular T-joint used in offshore structures, which was used for real-world data collection in this work.

the number of required points needed to describe them varies. The groove corners are natural choices for the desired subset of points. These corner points are often found through analysis of the first- and second derivatives using finite differences methods [3]–[6], by utilizing RANSAC search [7]–[9], or through combinations of the two [10]. Methods like template matching can also be used for weld groove parametrization [11], which perform well when some parts of the data are missing. Weld grooves are usually made by machining or grinding, making the groove surface reflective and leading to noisy data as a reflected laser line is triangulated. Therefore, feature extraction algorithms should be able to handle some data noise robustly. It is noted that the majority of methods are derived for processing the image data obtained directly from the camera of the structured light sensor [6], [9], [11]–[14], while others process the triangulated points, which is the final sensor output [3], [5], [7], [15]. The latter methods can be used with sensors where access to the raw image data is limited.

Neural networks have recently shown an ability to solve tasks from various domains. Convolutional Neural Networks (CNNs) have been used for feature extraction in the recent past [16]–[19]. However, all of these methods work at the camera image level. To the best of the authors' knowledge, CNNs have not yet been applied for weld groove feature extraction at the triangulated point output level.

In this work, we extend the results of [7] by proposing a CNN-based method for feature point extraction from triangulated point output of line laser scanners. The motivation for us-

<sup>1</sup>Dept. of Mechanical and Industrial Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway (email: oyvindmjolhus@gmail.com, olav.egeland@ntnu.no).

<sup>2</sup>Dept. of Production Technology, SINTEF Manufacturing, Trondheim, Norway (email: andrej.cibicik@sintef.no, eirik.njaastad@sintef.no).

The research presented in this paper has received funding from the Norwegian Research Council. The project is named Automated production of large steel structures, with project number 282106.

ing CNN compared to the previously used RANSAC algorithm lies in further reduction of computational time and extending the algorithm’s applicability to various types of grooves. In addition, we propose a methodology for generating synthetic training data in computer graphics software Blender, where accurate laser line modeling and data augmentation methods are designed to imitate naturally occurring noise, increasing the network robustness to noise caused by reflections of the laser line. This facilitates straightforward implementation of the proposed method for new types of weld grooves.

The rest of this work is organized as follows. In Section II, the background challenges of this work are presented. The proposed CNN-based method is given in Section III, while experimental verification and comparison are given in Section IV. The findings of the work are concluded in Section V.

## II. PROBLEM FORMULATION

Manual offline programming of a robotic welding system is a time consuming task, which is reasonable when a robot is welding the same mass produced product. In a case of welding large and bulky products, like ships, offshore floaters or substructures, relatively large geometric tolerances of stock steel products make every joint slightly geometrically different. This makes it impossible to use the same offline welding program and sensor feedback is required for adaptive welding robot path planning. One of the typical sensors for collecting weld groove geometric data is a line laser scanner, which returns a 2-dimensional (2D) array of points. This data has to be parametrized to a minimal set of parameters which can uniquely describe the weld groove shape and can be used for robot path planning. This task can be especially challenging when the dataset contains noise from the laser line reflections from the ground weld groove surfaces. In addition, data processing time is important as it might influence the efficiency of the whole process. Therefore, solution to the weld groove data parametrization problem is an important initial step towards more adaptive and flexible robotic welding systems.

The scope of this work is limited to grooves in tubular T-joints, however the methodology is general and can also be applied to other types of grooves.

## III. PROPOSED CNN-BASED WELD GROOVE FEATURE POINT EXTRACTION METHOD

We propose to train the CNN for weld groove feature extraction on data generated in a 3D graphics model. In the following section, we present the steps of the proposed approach.

### A. Synthetic Dataset Generation

1) *Model Setup and Data Generation:* One of the main challenges of successfully utilizing machine learning for a task is the need for large amounts of data. Data gathering and labelling are time-consuming and expensive without access to public datasets. One way to alleviate these concerns is by using synthetic data. Here, one avoids the need for physical objects



Fig. 2. Example of an entire scene consisting of a T-joint segment, an ambient background environment and a projected laser line (to the left). The appearance of the scene as seen from the camera of the virtual laser scanner (to the right).

to gather data, while additionally, it may remove the need to label data manually.

The open-source 3D modeling software Blender was used to create a synthetic dataset of various brace- and leg element configurations. Blender offers a built-in Python interpreter, allowing for the programmatical generation of numerous scenes showing the groove geometry of a sampling of various T-joints.

An example of such a scene can be seen in Fig. 2. An industrial laser scanner can be modeled by creating a virtual camera and a light source, where the shape and color of the projected light can be specified. We chose to emulate the Micro-epsilon scanCONTROL 2610-100 laser scanner in this work. Therefore, the virtual light source was set to project a single red laser line. In order to make the laser appear more realistic, different object parameters can be modified until it resembles a real-life laser scanner. This shows one of the challenges of using synthetic data: many modeling decisions must be made using heuristics.

Physically-Based Rendering (PBR) materials are available in the form of image texture maps. These can be projected onto the 3D models in the scene, assigning realistic material properties. This makes light interact with the models in a realistic manner.

During the acquisition of synthetic data, the emulated laser scanner is moved in discrete steps along the weld groove. For each step, three images are collected: one rendered image to be used for the training, one ideal image arranged for straightforward creation of training data labels, and one high-dynamic range, multi-channel raster image (EXR), which additionally contains the positional information of the image content as seen from the global reference frame. The scene used to create the ideal image differs by using no ambient lighting and assigning diffuse material properties to the elements. This results in light being reflected off the scene content equally in all directions, making the acquired image noise-free.

2) *Point Extraction:* Commercial laser sensors vary in the characteristics of their output. The sensor emulated in this work (Section III-A1) returns a set of 640 points in  $\mathbb{R}^2$ , given in the sensor frame coordinates. The synthetic data, therefore,

had to be processed to produce similar output.

Given a set of training data images, which is a rendered image with its corresponding ideal image, a number of steps are required to turn the images into point clouds. For the rendered image, typical image pre-processing techniques are applied before extracting the centre line of the laser. Using the open source library OpenCV, images are converted into the hue, saturation, and brightness color representation (HSV), followed by filtering with a median filter that replaces each pixel with the median of neighbouring pixels. Then, a binary mask with non-zero values only at areas where the hue, saturation and brightness constitute a red color is applied to zero out non-red areas. Then, the images are processed by a closing followed by an opening operation, which are combinations of dilating and eroding morphology operations. The morphology operations close gaps within the line and remove noise while thinning out the laser line. Finally, the images are converted into a grayscale format. There is no need for further processing for the ideal images, as they are generated to contain only the laser line before saving in grayscale.

Following the steps above, a *grey-gravity* method [20] is used for finding the weighted centre of the laser line in its thickness direction by aggregating the pixels with the highest intensities column-wise. When the centre line pixel coordinates have been extracted for both images, the position in the world reference frame is found by reading the values of the EXR image at the positions of the extracted pixel coordinates. Letting  $\tilde{\mathbf{p}}_j^w \in \mathbb{R}^4$  denote an obtained homogeneous point given in the world reference frame and  $\mathbf{T}_w^s \in \text{SE}(3)$  is the transformation from the laser scanner reference frame to the world reference frame, the homogeneous point as seen from the scanner,  $\tilde{\mathbf{p}}_j^s \in \mathbb{R}^4$ , is then calculated as  $\tilde{\mathbf{p}}_j^s = \mathbf{T}_w^s \tilde{\mathbf{p}}_j^w$ , where  $\mathbf{T}_w^s$  is obtained from Blender. Then the Euclidean point  $\mathbf{p}_j^s \in \mathbb{R}^2$  is obtained by removing the two last coordinates from  $\tilde{\mathbf{p}}_j^s$ . It is noted that the  $z$  coordinate of  $\tilde{\mathbf{p}}_j^s$  is always equal to zero.

3) *Data Labeling*: For the automatic labelling of data, we make use of the fact that methods based on derivative analysis have yielded favourable results on relatively noise-free data [3]. The previously mentioned ideal image is devoid of noise, and therefore, a simple corner detection algorithm based on identifying the peaks of the second derivatives is developed and used on this image to find the labels to be used with the training data.

## B. Feature Point Extraction

1) *Neural Network Setup*: A CNN is set up and used for the feature extraction. This popular class of artificial neural networks is favored due to its performance and computational efficiency. With the input data being a set of 2D points, where the  $x$  coordinates are sorted along the profile, it bears some resemblance to data typically seen in signal processing, with the  $x$  values being analogous to the time domain of a typical 1D signal. Therefore, 1D convolutional layers, common when

TABLE I  
NEURAL NETWORK ARCHITECTURE

Layer	Output shape	Param. #
conv1( <i>Conv1D</i> )	(Batch, 8, 128)	248
bn1( <i>BatchNorm1D</i> )	(Batch, 8, 128)	16
ReLU		
conv2( <i>Conv1D</i> )	(Batch, 16, 128)	400
bn2( <i>BatchNorm1D</i> )	(Batch, 16, 128)	32
ReLU		
MaxPool1D	(Batch, 16, 64)	
conv3( <i>Conv1D</i> )	(Batch, 32, 64)	1 568
bn3( <i>BatchNorm1D</i> )	(Batch, 32, 64)	64
ReLU		
MaxPool1D	(Batch, 32, 32)	
conv4( <i>Conv1D</i> )	(Batch, 64, 32)	6 208
bn4( <i>BatchNorm1D</i> )	(Batch, 64, 32)	128
ReLU		
MaxPool1D	(Batch, 64, 16)	
conv5( <i>Conv1D</i> )	(Batch, 128, 16)	24 704
bn5( <i>BatchNorm1D</i> )	(Batch, 128, 16)	256
ReLU		
MaxPool1D	(Batch, 128, 8)	
conv6( <i>Conv1D</i> )	(Batch, 256, 8)	98 560
bn6( <i>BatchNorm1D</i> )	(Batch, 256, 8)	512
ReLU		
MaxPool1D	(Batch, 256, 4)	
fc1( <i>Linear</i> )	(Batch, 100)	102 500
ReLU		
fc2( <i>Linear</i> )	(Batch, 10)	1 010

working with time-varying signals, were used in place of the more common 2D layers.

With execution time being an essential factor in the realization of adaptive robotic welding, the CNN was designed with the goal of being as efficient as possible. The architecture was kept simple, using the well-known Rectified Linear Unit (ReLU) activation function with intermediate Batch Normalization (BatchNorm) layers. The architecture can be seen summarized in Table I. The network outputs ten scalars for each input, corresponding to the  $x$ - and  $y$ -values of the five feature points. A custom loss function was implemented, calculating the average Euclidean distance between the estimated points and their corresponding labels. We thus formulated the problem as a regression task, seeking to minimize the Euclidean distance between the output and the labels.

When training the network, 10% of the dataset images were set aside for validation to monitor the performance on unseen data. The Adam optimizer was used for training, and the learning rate was gradually reduced by halving it every 200 epochs. An extensive data augmentation regime was used where the point clouds had a sampled set of augmentations applied to them, such as, e.g., a stochastic affine transformation or the addition of noise to random segments of the point cloud. The network was trained for 1200 epochs, taking around one hour on a GTX 3090 GPU, a point beyond which additional training leads to overfitting.

2) *Corner Correction Algorithm*: A least-squares line fitting algorithm can be used to increase the precision of the initial estimate of feature points. In general, such algorithms tend to improve the localization of feature points, especially for cases with rounded weld profile corners. In this work, we

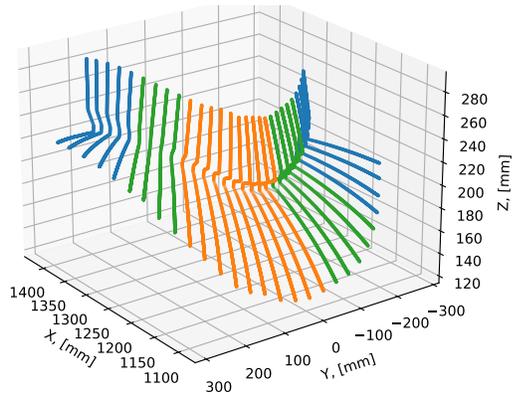


Fig. 3. A section of the real-world dataset used in the experimental verification. The groove geometry continuously varies along the circumference of the joint. The different colors illustrate areas with similar groove geometry.

implemented an iterative corner correction procedure similar to the one proposed in [7] for the final adjustment of the feature point estimates obtained by the CNN. It is noted that the feature point positions obtained by CNN should already be close to the true values. Otherwise, the corner correction algorithm forces initial estimates to diverge from the true values.

#### IV. EXPERIMENTAL STUDY AND COMPARISON

The performance of the feature extraction method was evaluated using real-world data collected in [7] and [8]. The method was implemented using Python and the deep learning framework Pytorch and was evaluated in terms of accuracy and execution time compared to the results obtained by the algorithm used in [7], [8].

The first set of real-world test data was small [7] with only a few samples but contained a substantial amount of reflection noise, making it particularly important for evaluation. This dataset was collected by intentionally creating conditions for high noise levels. The second set [8], collected from an actual large welded tubular joint, contained over 650 point clouds with less noise, see Fig. 1 and Fig. 3. This dataset was collected under realistic conditions and represents the quality of data typically found in an industrial setup.

The scope of the experimental evaluations was limited to stub joints with existing root welds of widths varying from 3-10 mm, which is a common size of the root weld for T-joints made with tubular sections with up to 1000 mm in diameter.

##### A. Results and Discussion

1) *Computational Efficiency*: The method was evaluated using a laptop with a GTX 1060 GPU, a quad-core Intel i7 CPU, and 16 GB of RAM. Even when using a relatively low-end GPU such as this one, the inference time of the neural network was approximately 1.0 ms when averaging over 300 individual forward passes. When using the CPU, this only increases by a small amount, needing about 1.2 ms, averaging over 300 forward passes. The fast inference time on the CPU

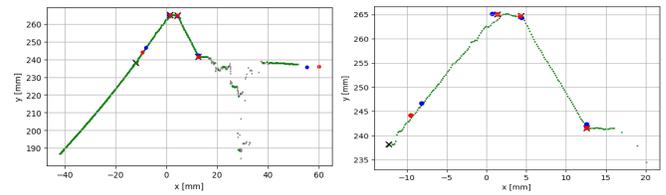


Fig. 4. The results when testing on samples from the noisy dataset [7], with a close-up shown to the right. Blue points are the original estimates of the CNN, red points are outputs after the corner correction algorithm, and the black cross marks are the results obtained in [7]. The gray points are identified as noise during the corner correction.

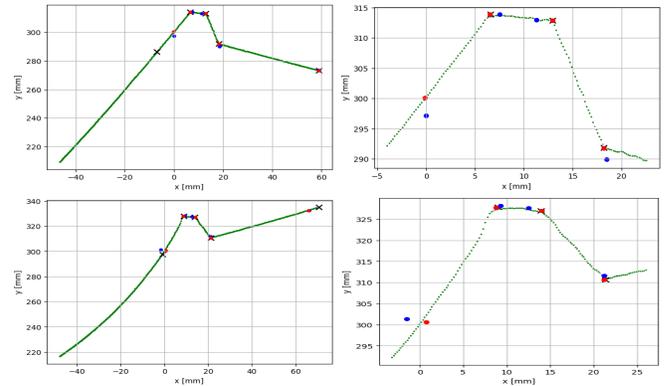


Fig. 5. Results of testing the method on samples from the dataset [8], with close-ups shown to the right. Two cases with different curvatures of the leg element are presented. The black cross marks are the outputs obtained in the previous work [8].

was likely due to the network itself being small and the low dimensionality of the input.

The corner correction algorithm dominated the total execution time of the method. Here, the execution time had a higher variance when processing different point clouds, depending on the accuracy of the initial estimates. The average execution time of the corner correction algorithm was approximately 5.4 ms, meaning that the total feature point extraction time was about 6-7 ms. The method thus remained at least five times faster than the one proposed in [7] when considering execution time.

2) *Feature Extraction Accuracy*: Both real-world datasets were used for the evaluation of accuracy. However, the second and larger dataset was used for statistical accuracy analysis due to its dominating size. It is noted that the accuracy was evaluated for inner corner points only.

An example of the feature point output for a very noisy dataset is given in Fig. 4. The proposed method was able to identify feature points of the profile accurately. The figure illustrates how the subsequent corner correction algorithm successfully improved the initial estimates.

Two typical examples of the weld groove profile data from the second dataset are shown in Fig. 5, where samples with different geometry and curvature of the leg element are presented. The corner points found by the method in [7] are included for comparison. The two methods produce equivalent

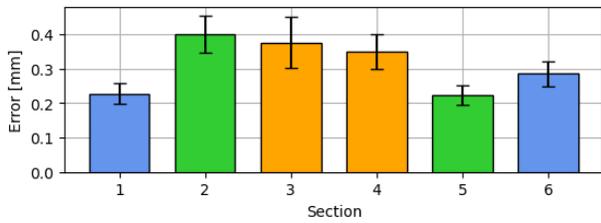


Fig. 6. The mean (shown with bars) and standard deviations (shown with vertical lines) of the distances between the proposed method and the one from [7]. Similar colored bars correspond to sections that are similar due to the symmetry of the joint.

results when considering accuracy, with the main difference stemming from how the corner point along the surface of the leg element is defined.

The profiles shown in Fig. 5 demonstrate how different the weld groove geometry can be along the groove of a T-joint. Most notably, the curvature of the line yielded by the leg element will vary, as will the angle between the brace and leg. Therefore, evaluation of accuracy was done separately for six consecutive sections of a T-joint scan of approximately 180 deg (see Fig. 3), where each section represents similar groove profile geometry. The performance of the proposed method for the six defined sections is shown in Fig. 6, where the graph shows the error and standard deviation of the feature point estimates compared to the results in [7], [8]. The best average precision was obtained for the first and last side sections, where the leg element has little curvature. The mean error for both sections was 0.26 mm. This was expected since most synthetic training data was generated using small leg element curvatures. However, the mean error for all other sections remained below 0.4 mm, which is well within the practical precision necessary for robotic welding of large tubular T-joints for offshore structures.

## V. CONCLUSION

In this work, we have suggested a feature point extraction algorithm for weld groove scan data for the case where a line laser scanner is combined with an industrial robot. The method used a CNN, which was trained using generated synthetic data. This way, the efforts required for training the network were significantly reduced. In addition, an iterative feature point correction procedure was implemented for accuracy improvement.

A case of a large tubular T-joint with varying weld groove geometry was investigated. The proposed algorithm yielded execution times at least five times smaller than the previous work while achieving similar precision. Low execution time is essential in applications where the feature points are used for online feedback control of a welding robot.

## REFERENCES

[1] L. Yang, Y. Liu, and J. Peng, "Advances techniques of the structured light sensing in intelligent welding robots: a review," *The International Journal of Advanced Manufacturing Technology*, vol. 110, no. 3, pp. 1027–1046, 2020.

[2] T. Lei, Y. Rong, H. Wang, Y. Huang, and M. Li, "A review of vision-aided robotic welding," *Computers in Industry*, vol. 123, p. 103326, 2020.

[3] G. Zhang, Y. Zhang, S. Tuo, Z. Hou, W. Yang, Z. Xu, Y. Wu, H. Yuan, and K. Shin, "A novel seam tracking technique with a four-step method and experimental investigation of robotic welding oriented to complex welding seam," *Sensors*, vol. 21, no. 9, p. 3067, 2021.

[4] D. Chang, D. Son, J. Lee, D. Lee, T.-w. Kim, K.-Y. Lee, and J. Kim, "A new seam-tracking algorithm through characteristic-point detection for a portable welding robot," *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 1, pp. 1–13, 2012.

[5] R. Manorathna, P. Phairatt, P. Ogun, T. Widjanarko, M. Chamberlain, L. Justham, S. Marimuthu, and M. R. Jackson, "Feature extraction and tracking of a weld joint for adaptive robotic welding," in *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*. IEEE, 2014, pp. 1368–1372.

[6] X. Lü, D. Gu, Y. Wang, Y. Qu, C. Qin, and F. Huang, "Feature extraction of welding seam image based on laser vision," *IEEE Sensors Journal*, vol. 18, no. 11, pp. 4715–4724, 2018.

[7] A. Cibicik, L. Tingelstad, and O. Egeland, "Laser scanning and parametrization of weld grooves with reflective surfaces," *Sensors*, vol. 21, no. 14, p. 4791, 2021.

[8] A. Cibicik, E. B. Njaastad, L. Tingelstad, and O. Egeland, "Robotic weld groove scanning for large tubular T-joints using a line laser sensor," *The International Journal of Advanced Manufacturing Technology*, pp. 1–14, 2022.

[9] N. Wang, K. Zhong, X. Shi, and X. Zhang, "A robust weld seam recognition method under heavy noise based on structured-light vision," *Robotics and Computer-Integrated Manufacturing*, vol. 61, p. 101821, 2020.

[10] Y. Han, J. Fan, and X. Yang, "A structured light vision sensor for on-line weld bead measurement and weld quality inspection," *The International Journal of Advanced Manufacturing Technology*, vol. 106, no. 5, pp. 2065–2078, 2020.

[11] X. Li, X. Li, S. S. Ge, M. O. Khyam, and C. Luo, "Automatic welding seam tracking and identification," *IEEE Transactions on industrial electronics*, vol. 64, no. 9, pp. 7261–7271, 2017.

[12] H. Chen, W. Liu, L. Huang, G. Xing, M. Wang, and H. Sun, "The decoupling visual feature extraction of dynamic three-dimensional v-type seam for gantry welding robot," *The International Journal of Advanced Manufacturing Technology*, vol. 80, no. 9, pp. 1741–1749, 2015.

[13] Q.-Q. Wu, J.-P. Lee, M.-H. Park, B.-J. Jin, D.-H. Kim, C.-K. Park, and I.-S. Kim, "A study on the modified hough algorithm for image processing in weld seam tracking," *Journal of Mechanical Science and Technology*, vol. 29, no. 11, pp. 4859–4865, 2015.

[14] J. Marco-Rider, A. Cibicik, and O. Egeland, "Polarization image laser line extraction methods for reflective metal surfaces," *IEEE Sensors Journal*, 2022.

[15] J. Ratava, S. Penttilä, M. Lohtander, and P. Kah, "Optical measurement of groove geometry," *Procedia Manufacturing*, vol. 25, pp. 111–117, 2018.

[16] Y. Zou, J. Li, X. Chen, and R. Lan, "Learning siamese networks for laser vision seam tracking," *JOSA A*, vol. 35, no. 11, pp. 1805–1813, 2018.

[17] Y. Zou, R. Lan, X. Wei, and J. Chen, "Robust seam tracking via a deep learning framework combining tracking and detection," *Applied Optics*, vol. 59, no. 14, pp. 4321–4331, 2020.

[18] K. Wu, T. Wang, J. He, Y. Liu, and Z. Jia, "Autonomous seam recognition and feature extraction for multi-pass welding based on laser stripe edge guidance network," *The International Journal of Advanced Manufacturing Technology*, vol. 111, no. 9, pp. 2719–2731, 2020.

[19] Y. Zou, T. Chen, X. Chen, and J. Li, "Robotic seam tracking system combining convolution filter and deep reinforcement learning," *Mechanical Systems and Signal Processing*, vol. 165, p. 108372, 2022.

[20] Y. Li, J. Zhou, F. Huang, and L. Liu, "Sub-pixel extraction of laser stripe center using an improved gray-gravity method," *Sensors*, vol. 17, no. 4, p. 814, 2017.