Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-31 (2016) 73–78

Automatic Touch-Up of Welding Paths Using 3D Vision

Eirik B. Njaastad [*] **Olav Egeland** [**]

[*] *Norwegian University of Science and Technology, Department of Production and Quality Engineering, S.P. Andersens veg 5, NO-7491 Trondheim, Norway (e-mail: eirik.njaastad@ ntnu.no).*
[**] *Norwegian University of Science and Technology, Department of Production and Quality Engineering, S.P. Andersens veg 5, NO-7491 Trondheim, Norway (e-mail: olav.egeland@ ntnu.no).)*

Abstract: This paper presents a system for automatic robotic welding based on offline programming using CAD data. The welding paths are corrected before execution with 3D vision where the 3D image is aligned with the CAD model of the workpiece to be welded. The system is successfully validated in experiments, and the results are presented in the paper.

Keywords: Computer Vision, Flexible Manufacturing Systems, CAD/CAM/CAE

1. INTRODUCTION

Robotic welding gives good repeatability of the welding path trajectories, as robots performs well in structured environments. Due to tolerances in the workpiece geometries and uncertainty in workpiece position and orientation, the desired welding path will vary from one workpiece to another. In contrast to manual welders, the robot is not able to modify the welding path continuously by itself. To make robotic welding systems more flexible in dealing with varying workpieces, external inputs such as computer vision must be introduced.

Robot programming with a teach pendant is often characterized as tedious and time-consuming, with a lack of intuitive tools for interaction. Many small and medium-sized enterprises (SMEs) are not using robots in their facilities because the configuration and programming process of this type of equipment is time-consuming and requires workers with a high level of expertise in the field (Neto and Mendes, 2013). Thus, there is a great potential for improvements in programming that enable cost-effective production of high quality welds at shorter cycle times.

In Offline Robot Programming (OLP), the robot motions are programmed without using a real robot. CAD systems are used to model the particular robot, workpiece, tool, and workspace. The models are then used to simulate robot tasks and path planning, and to generate programs that are downloaded and executed by the robot controller. Before the program can be executed, it is usually necessary to perform verification and small changes to the program (Pan et al., 2010). This part of the process is called program *touch-up* and is typically performed as *Lead-Through* or *Walk-Through* programming. By current methods, the robot programs is about 75 percent completed before manual touch-up (Pires et al., 2006).

Conventional approaches for program *touch-up* includes using a set of calibration points within the robot cell, or compensating for the discrepancies through the use of sensors on the real robot. Some adaptation strategies focus on real-time tracking of the welding seams based on 2D laser sensors (Manorathna et al., 2014), (Fang et al., 2011), or by through-the-arc sensing of fluctuations of welding current or voltage occurring when the welding torch is advanced in a weaving pattern (Jeong et al., 2001). Force feedback or tactile sensing can be used to locate the start point of the welding seams, by determining contact points between the welding torch and workpiece (Sanders et al., 2010). These approaches has to be reconfigured for each specific workpiece and is often best suited for simple workpiece geometries.

Other approaches focus on precise localization of the workpiece before offline programming of the robot motions (Pan et al., 2010). The actual workpiece pose can be registered and calibrated to correspond to the CAD model by using optical sensors (Rajaraman et al., 2013), (Dietz et al., 2012). In some approaches, computer vision is used for approximation and reconstruction of the actual workpiece geometry and then automatic generation of robot programs for each new workpiece (Dinham and Fang, 2012).

Enhanced use of computer vision for detecting and measuring the physical location and orientation of workpieces to be welded could improve the efficiency of welding processes by offline programmed robots, and the required level of competence for using a robotic welding system could potentially be lowered. This will allow many more enterprises to benefit from robotic welding systems.

RGB cameras are available in a wide range and at low cost. However, developing robust and efficient vision guided robotic systems can be a challenging task to accomplish when using 2D images that inherently only captures a projection of the 3D world. The recent arrival of Time-of-Flight (ToF) depth cameras and the even more recent introduction of the Microsoft Kinect^{TM} depth camera has

2405-8963 © 2016, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.12.164

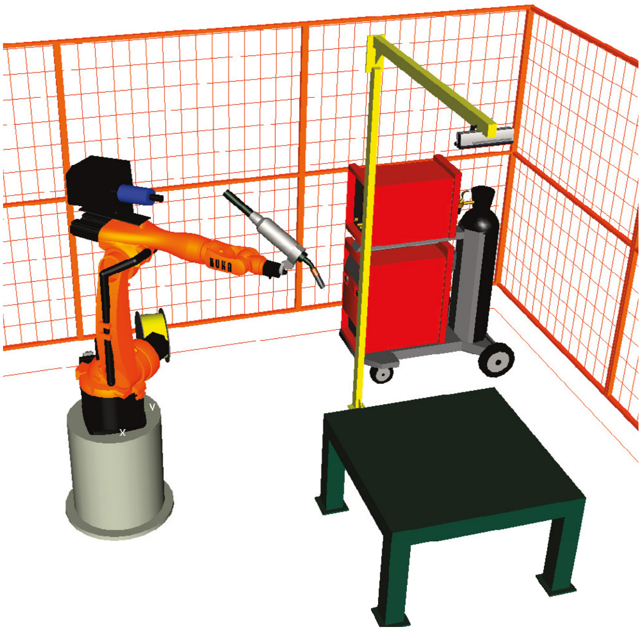

Fig. 1. Graphical model of the test cell layout. The Kinect camera is mounted over the worktable, with its field of view oriented towards the table.

made 3D data streams at video rate widely available. This enables new methods for helping robots to make more useful decisions.

This paper studies the possibility of using a low cost 3D camera and CAD model data for correcting offline programmed welding path trajectories. An experimental system has been developed and implemented, and a series of tests have been performed to evaluate the performance of the system. The paper is organized as follows: The second section of this paper presents the necessary steps used in the computer vision part of this work. The third section describes the implementation of the system, with focus on the information flow and an example of a pose estimation performed by the system is presented. Finally, the last section is a discussion on the results obtained.

2. OBJECT POSE ESTIMATION FROM CAD MODEL AND 3D-CAMERA DATA

2.1 Point Cloud Acquisition

The Microsoft Kinect $^{\text{TM}}$is an optical time-of-flight (ToF) camera which measures the depth of a scene by illuminating the scene with a modulated light source, and observing the reflected light (Payne et al., 2014). The Kinect for Xbox One is the second generation of sensor input devices developed for the Microsoft Xbox video game console systems, comprising state of the art depth sensing technology at a low cost. This version of the 3D camera includes a 512 x 424 pixel wide-angle ToF camera. The ToF camera is of the Continuous Wave Modulation type, with 70° horizontal and 60° vertical field of view.

Calibration of the 3D camera can be performed by using the Camera Calibration Toolbox for MATLAB (Bouguet, 2004), an implementation of the method proposed by Zhang (1999).

2.2 Point Cloud Processing

The acquired depth data is processed in order to enhance the quality of the data obtained in the point cloud acquisition step. When acquired, the point clouds are often degraded due to distortion and noise in the camera system. Another problem is the massive amount of data captured in each point cloud, which can greatly reduce the effect of recognition and alignment algorithms. Down-sampling, smoothing, segmentation, and estimation of local surface geometry are examples of operations necessary to make the data in point clouds more useful.

2.3 Object Alignment

For point clouds, object alignment is the problem of finding correct point correspondences in a given dataset, and estimating transformations that can rotate and translate each individual dataset into a consistent coordinate framework.

One of the strategies for aligning point clouds is to search for the right transformation by estimating correspondences, then estimate a transformation given a correspondence, and repeating (Forsyth and Ponce, 2012). Such approaches can be classified as local optimization methods, and in this category a widely used method is the *Iterative Closest Point* (ICP) algorithm (Rusu et al., 2009). Development of an offline programmed robotic welding sequence includes establishing the pose of a simulated representation (i.e., a CAD model) of the object to be welded. This pose usually provides a good initial guess of the true physical object pose, and serves as a good starting point for object aligning by local approaches. The initial alignment is relatively easy to obtain for applications described in this work. ICP, which is considered a fine tuning alignment method (Rusu and Cousins, 2011), then emerges as a good alternative for object alignment. The ICP algorithm was introduced by Chen and Medioni (1991) and Besl and McKay (1992) in the early 90s, but many variations on the basic ICP concept have later been introduced.

3. SYSTEM IMPLEMENTATION

3.1 Robot Cell

A robot cell was set up for experimental evaluation of the solution. The robot cell layout is illustrated in Fig. 1. The robotic welding system had a KUKA KR 5 robot and a Fronius TransSteel 5000 welding machine. The welding machine was mounted on the KR 5 robot manipulator and communicates with the robot controller. The welding machine was set up for Metal Active Gas (MAG) welding.

3.2 Information Flow

The system combines data from the depth camera with prior knowledge from a CAD model and an offline programmed welding sequence to determine the location and orientation of the welding path trajectories. Existing KUKA software was the basis for distributing information in the system. A self-developed C++ application served as the interface between the Kinect camera and the robot, and handled the different algorithms and calculations. Fig. 2 gives an overview of the information flow in the system.

The simulated 3D position and orientation of the component to be welded are the required input parameters for the object alignment. It represents the system's initial guess for the pose of the physical component, and any deviations are measured relative to this pose. The input parameters were represented in terms of (X, Y, Z, R_x, R_y, R_z) values.

The simulation suite called *KUKA.Sim* was used for both building a 3D model of the robot cell and for programming the welding sequences. In addition to information about planned positions, velocities, and accelerations, the programmed sequences also included welding parameters such as welding current and weaving data for each weld. The 3D points in the offline programmed robot welding sequence originated from a coordinate frame located on the worktable. If the system detected deviations from the programmed and simulated component pose, the points were transformed into the new location and orientation. This transformation was performed by the robot controller at program runtime, by transforming the origin of the coordinate frame. The new pose estimation was represented in terms of $(X', Y', Z', R'_x, R'_y, R'_z)$ values.

The depth stream from the Kinect 3D camera was running at 30 frames per second. This stream was captured and converted into point clouds by using the standard driver provided by Microsoft. In order to align the CAD model with the captured point clouds, the model was converted to a point cloud by sampling a specified number of points on all surfaces of the model. In the developed system, CAD files of the STereoLithography (STL) format was used.

Fig. 3 shows a captured point cloud and the point cloud representation of the CAD model before and after alignment. Object alignment was performed by using the ICP algorithm with Levenberg-Marquardt optimization. When the algorithm has converged to a solution, it returns a 4×4 transformation matrix and the correction is transmitted to the robot controller. The commands are sent to the robot as cartesian corrections. The KUKA KR5 robot system describes orientations by Tait-Bryan angles, using the Yaw Pitch Roll (Z, Y, X) composite rotation convention.

4. RESULTS

4.1 System Performance

The system has been evaluated by performing welding operations at various component positions and orientations. A fillet weld was performed in both horizontal and vertical direction for various poses. In Fig. 4, the two welding paths are marked on a sample component. The horizontal welding joint is performed by following an edge through a 90 degree corner, and therefore runs in both directions of the robot XY plane. The vertical welding joint is carried out almost straight down along the robot Z axis.

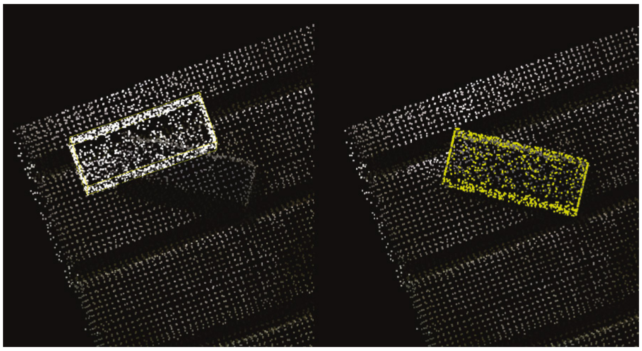

Fig. 3. Object alignment by Iterative Closest Point (ICP). The left side shows the acquired point cloud and the point cloud representation of the CAD model (white points). The right side shows how ICP aligns the CAD model to the acquired data (yellow points).

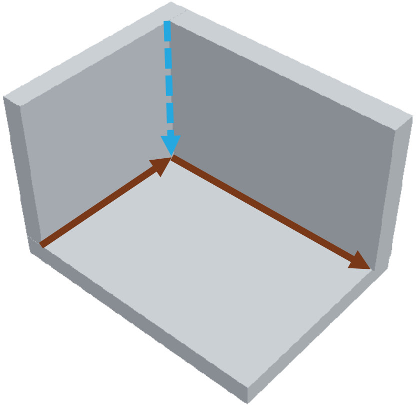

Fig. 4. The welding path trajectories used for testing the system performance marked on a sample component. The fillet weld in horizontal position is illustrated by brown, solid arrows. The fillet weld in vertical position is illustrated by a blue, dashed arrow.

In all tests, the performance has been evaluated by comparing the offline programmed welding path to the corrected welding path from the developed system, and a manually optimized welding path. The manually optimized path was made by teaching the path with the *walk-through* robot programming method. The measured deviations were thus found by using the robot end-effector, i.e., the tip of the welding electrode.

Case I: No Object Alignment As a reference for the comparisons, the system was first tested without corrections from the 3D camera. This corresponds to normal offline robot programming, where welding sequences are programmed based on pre-measured or assumed values for the component location and orientation. The results achieved from this standard approach were for the most part unfinished welding programs. The precision obtained were not good enough to perform welding directly.

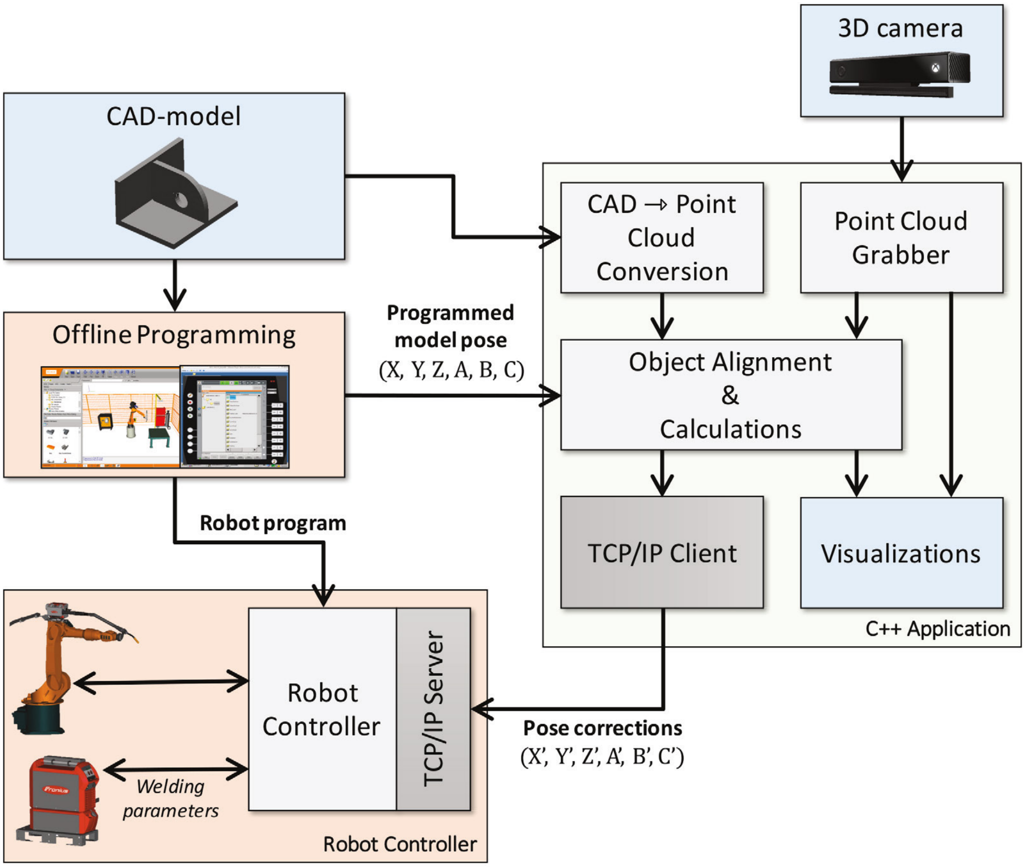

Fig. 2. Information flow of the developed system. A CAD model and data from a Kinect 3D camera are the inputs to the system. The CAD model is used for offline programming of the welding sequence, and as input to the object alignment process. The developed C++ application estimates a corrected component pose, and communicates the pose to the robot controller. In the illustration, the activities handled by the C++ application is shown in the rightmost (purple) box. Finally, the offline programmed welding paths are performed by the robot, based on the corrected component pose.

Case II: 2D Object Alignment. The next test included object alignments from the system, but was performed with limitations in the estimation of the object pose. The 3D camera was placed over the worktable with its field of view oriented towards the table. Because the camera Z axis is perpendicular to a coordinate system on the table surface, a simple 2D rigid transformation (X, Y, R_z) could be estimated as a first step.

When comparing the reference test without corrections with the results from the 2D object alignment, the latter showed clearly improved welding paths. As shown in Fig. 5, the adjusted trajectories are close to those of the desired welding paths. The accuracy in Z direction is equivalent of that achieved in the reference test.

Case III: 3D Object Alignment In this case, a full 3D transformation (X, Y, Z, R_x, R_y, R_z) is estimated. The offline programmed and resulting transformed welding paths are illustrated in Fig. 6. Compared to *Case I and II*, this solution performs similar to the 2D object alignment for estimations in X and Y directions. The full 3D alignment also has good performance for estimations in Z direction and for rotations.

The absolute errors of the welding path trajectories in Case I to III are shown in Fig. 7. For the solution in Case III, a mean absolute error of approximately 2.4 mm with a maximum of approximately 5.7 mm was achieved. This is not a sufficient result for all applications, but it is an acceptable deviation for many welding applications and promising for future work.

5. CONCLUSIONS

A 3D computer vision solution was developed in order to improve the process of offline programming a welding robot, by estimating a corrected pose of the object to be welded. The results were demonstrated by programming and welding a series of welding path trajectories at various component positions and orientations

The results show small variations in the corrected object pose estimation. A mean absolute error of approximately 2.4 mm with a maximum of approximately 5.7 mm was achieved.

ACKNOWLEDGEMENTS

This work was supported by Brunvoll AS, Oshaug Metall AS, and the Norwegian Research Council under the EFFEKT project.

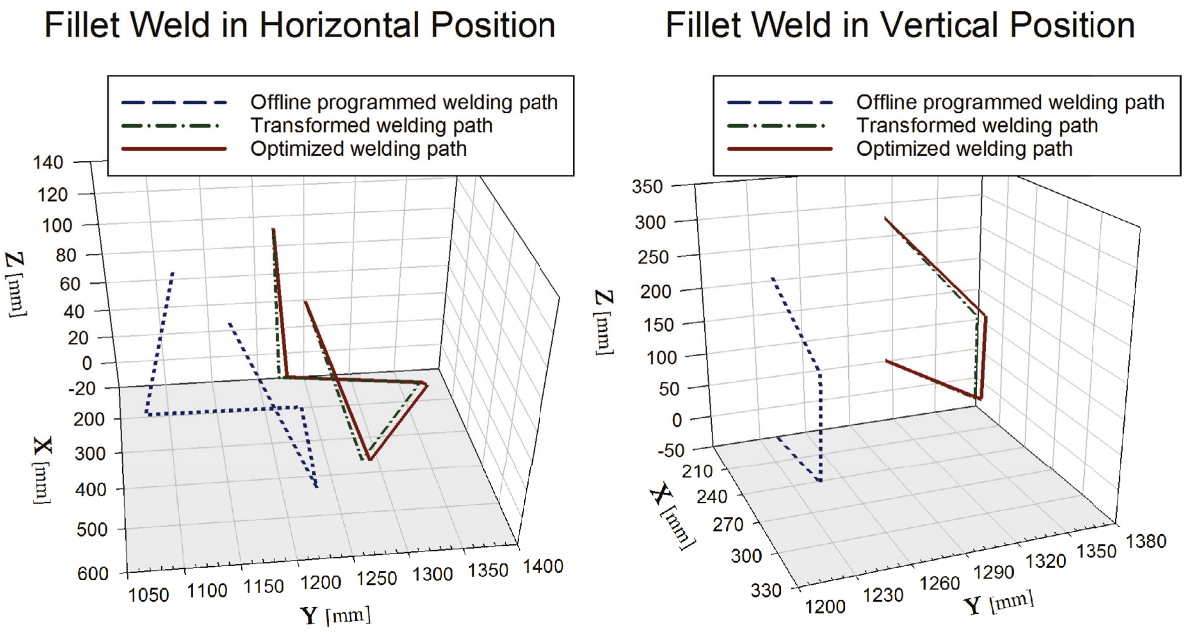

Fig. 5. In Case II, the robot executed the welding path corrected by 2D (X, Y, R_z) object alignment (dot-dashed, green line). In addition, the robot executed a welding path that was manually optimized (solid, red line). The horizontal fillet weld is shown to the left, and vertical fillet weld to the right. The offline programmed path is shown for both cases by the dashed, blue line.

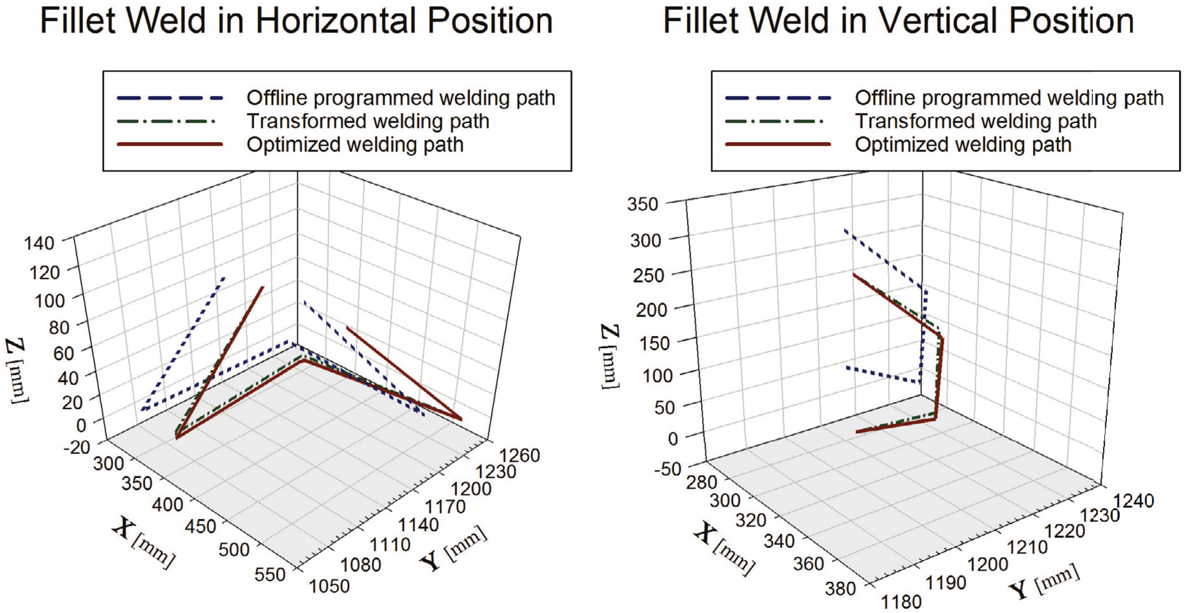

Fig. 6. In Case III, the robot executed the welding path corrected by 3D (X, Y, Z, R_x, R_y, R_z) object alignment (dot-dashed, green line). In addition, the robot executed a welding path that was manually optimized (solid, red line). The horizontal fillet weld is shown to the left, and vertical fillet weld to the right. The offline programmed path is shown for both cases by the dashed, blue line. The deviation between the two welding paths demonstrated the quality of the automatic method.

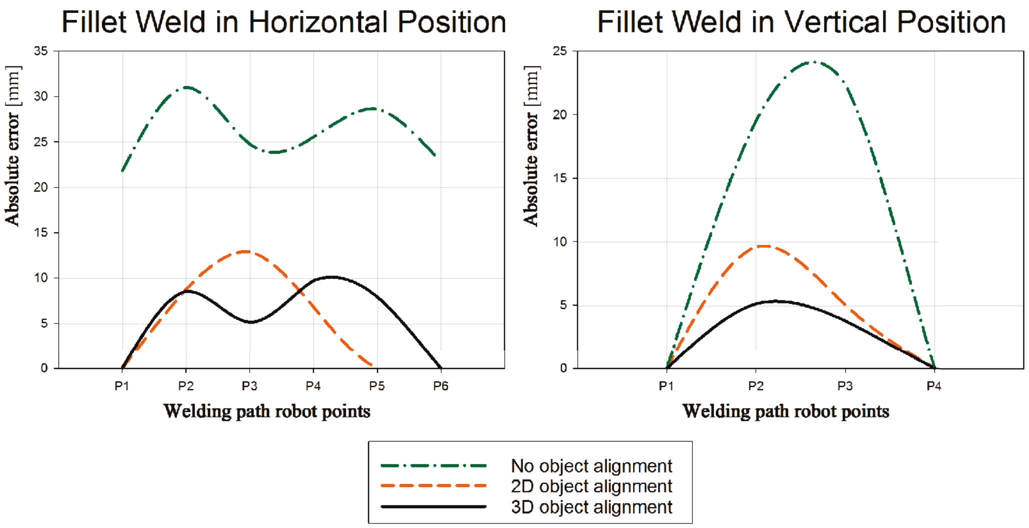

Fig. 7. Absolute error observed for the welding welding path trajectories in Case I to III. The horizontal fillet weld is shown to the left and vertical fillet weld to the right, where P1 to P6 is the actual programmed 3D poses for the robot manipulator. The errors were found by comparing the offline programmed and corrected welding paths from the developed system to an manually optimized welding path.

REFERENCES

Besl, P.J. and McKay, N.D. (1992). Method for registration of 3-d shapes. In *Robotics-DL tentative*, 586–606. International Society for Optics and Photonics.

Bouguet, J.Y. (2004). Camera calibration toolbox for matlab.

Chen, Y. and Medioni, G. (1991). Object modeling by registration of multiple range images. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference*, 2724–2729. IEEE.

Dietz, T., Schneider, U., Barho, M., Oberer-Treitz, S., Drust, M., Hollmann, R., and Hägele, M. (2012). Programming system for efficient use of industrial robots for deburring in sme environments. In *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*, 1–6. VDE.

Dinham, M. and Fang, G. (2012). Weld seam detection using computer vision for robotic arc welding. In *2012 IEEE International Conference on Automation Science and Engineering (CASE)*, 771–776. IEEE.

Fang, Z., Xu, D., and Tan, M. (2011). A vision-based self-tuning fuzzy controller for fillet weld seam tracking. *IEEE/ASME Transactions on Mechatronics*, 16(3), 540–550.

Forsyth, D.A. and Ponce, J. (2012). *Computer Vision: A Modern Approach 2nd Ed.* Pearson Eduaction, Inc., 2 edition.

Jeong, S.K., Lee, G.Y., Lee, W.K., and Kim, S.B. (2001). Development of high speed rotating arc sensor and seam tracking controller for welding robots. In *Industrial Electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on*, volume 2, 845–850. IEEE.

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

Neto, P. and Mendes, N. (2013). Direct off-line robot programming via a common cad package. *Robotics and Autonomous Systems*, 61(8), 896–910.

Pan, Z., Polden, J., Larkin, N., Duin, S.V., and Norrish, J. (2010). Recent progress on programming methods for industrial robots. In *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*.

Payne, A., Daniel, A., Mehta, A., Thompson, B., Bamji, C.S., Snow, D., Oshima, H., Prather, L., Fenton, M., Kordus, L., et al. (2014). 7.6 a 512× 424 cmos 3d time-of-flight image sensor with multi-frequency photo-demodulation up to 130mhz and 2gs/s adc. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*, 134–135. IEEE.

Pires, J.N., Loureiro, A., and Bölmsjo, G. (2006). *Welding robots: technology, system issues and application.* Springer Science &amp; Business Media.

Rajaraman, M., Dawson-Haggerty, M., Shimada, K., and Bourne, D. (2013). Automated workpiece localization for robotic welding. In *2013 IEEE International Conference on Automation Science and Engineering (CASE)*, 681–686. IEEE.

Rusu, R.B., Blodow, N., and Beetz, M. (2009). Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, 3212–3217. IEEE.

Rusu, R.B. and Cousins, S. (2011). 3d is here: Point cloud library (pcl). In *ICRA*. IEEE.

Sanders, D.A., Lambert, G., Graham-Jones, J., Tewkesbury, G.E., Onuh, S., Ndzi, D., and Ross, C. (2010). A robotic welding system using image processing techniques and a cad model to provide information to a multi-intelligent decision module. *Assembly Automation*, 30(4), 323–332.

Zhang, Z. (1999). Flexible camera calibration by viewing a plane from unknown orientations. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, 666–673. IEEE.