E

**Proceedings of the ASME 2022**

**International Design Engineering Technical Conferences and   
Computers and Information in Engineering Conference**

**IDETC/CIE2022  
August 14-17, 2022, St. Louis, MO**

DETC2022-XXXX

**Automated Weld Path Generation Using Random Sample Consensus and Iterative Closest Point Workpiece Localization**

|  |  |  |
| --- | --- | --- |
| Tristan Hill  Tennessee Technological  University | Stephen Canfield  Tennessee Technological  University | Robert Shelton  Tennessee Technological  University |

Abstract

Jobs performed by small to medium enterprises (SMEs) are infrequently automated due to high setup costs and lack of technical expertise needed for robot training, however productivity and worker safety can be improved in SMEs with the use automated tooling. In a traditional automated manufacturing environment, tasks such a welding or painting are accomplished through execution of pre-programmed tool motions which rely on the location and orientation of the workpiece to be fixed and known. The lack of this spatial information is typically treated through positioning of the workpiece with respect to the robot arm using jigs or fixtures which are costly in initial setup and not easily modified. Further, the resulting toolpath associated with a desired task is typically defined through manual teaching resulting in a path appropriate for an individual job. For this reason, SMEs requiring variation in part geometry or arrangement are not commonly automated. This work presents a method for automated weld path generation for a 6DOF co-bot arm using random sample consensus (RANSAC) and iterative closest point (ICP) workpiece localization from LiDAR pointclouds. Scans from a low cost 2D LiDAR mounted to the co-bot arm are used to generate 3D pointclouds of the workspace scene with the Robot Operating System (ROS). The Point Cloud Library (PCL) is used to compare the generated pointcloud with a CAD model to produce a rigid transformation to localize the workpiece. The estimated pose of the workpiece with respect to a fixed frame is used offline to generate a weld path as series of tool poses. Two example welding processes in which a cylinder or rectangular tube is joined to a flat plate and two square tubes are joined through weldment are investigated and a physical implementation of the method is demonstrated using a 2D LiDAR mounted to a 6DOF co-bot carrying a MIG welding torch.

Keywords: robotics, automation, welding, ROS, PCL

Nomenclature

Place nomenclature section, if needed, here. Nomenclature should be given in a column, like this:

α alpha

β beta

1. INTRODUCTION

Small to medium enterprises perform manufacturing tasks associated with relatively low part volume and increased variation in assembly geometry as compared to jobs performed in large scale manufacturing environments. This type of manufacturing operation is infrequently automated due to high setup costs. However, productivity and worker safety can be improved in small to medium enterprises with automation such as robotic tooling.

In a traditional automated manufacturing environment, a task such a welding or painting is accomplished through the execution of pre-programmed tool motions which rely on the location and orientation of the workpiece to be fixed and known with respect to a global coordinate system. The need for spatial information is typically treated through positioning of the workpiece with respect to the robot arm using jigs or fixtures which are costly in initial setup and are not easily modified. In large scale production environments, this can be accomplished with dedicated infrastructure built into the environment such as moving jigs on assembly lines and other features available in a highly structured environment. Further, the resulting toolpath associated with a desired task is typically defined through manual teaching resulting in a path appropriate for an individual job. For this reason, operations performed by SMEs requiring lower volume manufacturing with variation in part geometry or arrangement are not commonly automated.

A method is presented in this paper is for automated weld path generation with a 6DOF co-bot arm using random sample consensus (RANSAC) and iterative closest point (ICP) workpiece localization implemented using the Robot Operating System (ROS) and the Point Cloud Library (PCL). Point Cloud Library provides an open-source C++ implementation of several 3D point cloud and image processing algorithms including: object recognition, filtering, feature estimation, surface reconstruction, registration, model fitting, and segmentation [5]. This library is an attractive research tool due to its stability, integration with ROS, and standard data sets that can be used for comparison repeatability in research.

**Literature Review**

Automated path planning for welding robots has been studied for decades [] and is an area of increasing interest as the use [9] and availability [] of robotics and sensing technology increases.

Seam tracking is a pre-dominate [] approach to the automation of welding. A vision system or other non-contact sensor detects and measures the location of seam to be welded, and this spatial information provides feedback for a control strategy to guide the tool through the required process. The detection and tracking of weld seams has been demonstrated through use of optical cameras with structured light [24, 35], stereo vision [23], and 3D LiDAR [33]. Shah et al provide a review of vision-based examples of weld seam identification, detections, and tracking [31].

Automation of the welding process has been shown using various feature detection techniques which inform the controller of spatial information of the weld seam without exposing the full pose of the workpiece with respect to the robot[]. In practice, the starting weld point (SWP) must be known for a weld to be successfully completed through seam tracking or other automatic controller. Autonomous determination of the start welding position (SWP) has been presented by Chen [32]. This is also shown in XX and YY.

Alternatively, the problem can be framed as a search for the location and orientation of the workpiece in a global sense. This information can be compared to a model to localize the weld seam with respect to the workpiece and robot. At a minimum, this requires the location of three points of the workpiece to be known[].

A process of extracting the geometry of a weld seam on flat parts lying on a 2D surface from stereo vision imagery is shown in [23]. Similarly, the discovery and localization workpieces to be joined by welding from 3D point cloud data is described by Rajaraman, et al. [28]. This paper will follow a similar process of part sensing to path planning.

Gao et al present a method for extraction of seam geometry in D-type weld based on point clouds [33]. This work presents a similar approach to weld seam localization through processing of point cloud data.

Schleth, Kuss, and Kraus provide an overview of the available literature related to workpiece registration in robot manufacuturing processes [20]. Kuss et al. present the problem of workpiece localization and detection of part alignment and shape variation in preparation for a manufacturing application such as welding or deburring. [20, 21,22]. Part alignment and its effect on the welding model used for automated process planning is investigated in [].

ANY PROS/CONS WITH THE CURRENT METHODS?

The work presented in the current paper follows [29,33] in the sense that the iterative closest point method is used to determine the rigid transformation representing the workpiece in a welding operation to be completed by a 6DOF robotic arm designed for automated manufacturing processes. The Pass-Through filter and Voxel down sampling used by Gao [33] is similar to the prefiltering routine used in this paper with the addition of the statistical filter to remove outliers in the point clouds. This paper will present examples of weld operations incorporating fillet-type welds on spatial (non flat) bodies.

The strategy presented in this work differs from that of Kuss et al in that the localization process is performed using global images of the entire scanned area including the table and clamps instead of isolated scans of the workpiece itself. A strategy for outlier detection and removal is implemented prior to image registration which improves the performance of the process.

A collection of these algorithms is used to locate, or register, a point cloud representing the workpiece in a point cloud of the working environment collected by a LiDAR scanner located on the robot. Once the known part is located with respect to a fixed frame, an automated weld path generation routine is used to plan a weld tool-path offline. Two example welding applications are presented. In the first, a square tube is joined to a flat plate through weldment. In the second, two square tubes are joined orthogonally to each other to form a tee. Simulations of both applications are investigated and a physical implementation of the method is demonstrated using a 2D LIDAR mounted to a 6DOF cobot carrying a MIG welding torch which can be seen below in Figure 1.

The primary algorithms and used in the approach presented are well known and accepted [2,12,6]. Random Sample consensus and iterative closest point have been thoroughly tested and documented. However, the iterative closet point method is limited by local minima convergence issues associated with poor initial search conditions [].

Several variants and improvements of the ICP algorithm have been shown [4,7,17] which aim to address the problems with optimization. Probabilistic methods have been applied to the image registration problem. The ICP problem is framed in probabilistic terms by Thurn et al to address the computation requirements of the algorithm [1]. An improved point cloud registration technique is presented in [34] which uses a hybrid approach with a genetic algorithm to predict the correspondences required by ICP.

[] Variants of registration algorithms which replace the optimization component of iterative closest point with machine learning algorithms are emerging [27,28,29]. TEASER applies These ml approaches to the image registration problem and show performance improvements as compared to [26]. Chen et al present an artificial neural network approach to 2D vision-based arc welding control [17]. This area has continued to develop as advances in machine learning and modern computer vision strategies become available.

Although there are many improvements and variations available, this paper will use classical point-to-point ICP as available with PCL []. The known issues associated with point cloud outliers and poor initial conditions are addressed in this paper by pre filtering and segmentation of the input point clouds prior to the use of ICP.

A picture containing wall, indoor

Description automatically generated

Figure 1

Environment sensing devices which generate 3D points are frequently used in the robotics industry, and improved sensors are being developed with the increased demand [5] for automation in manufacturing and transportation. A point cloud is a list of points in 3D space representing a physical object or collection of objects [6][8], and this data is generated through measurements from a sensing device such as a LiDAR or 3D camera. Widespread applications and research involving spatial data has led to the development of standard file types, storage containers, and libraries for efficiently processing of pointclouds [5]. Common programming languages (C++, Python, MATLAB) support integration of pointcloud data with various libraries (PCL, OpenCV) and software frameworks (ROS).

The geometrical data, or features, stored in a pointcloud (typically) contain the locations of the boundaries of a solid object. Features may also include point-normals which can be measured or inferred from the feature locations. Non-geometrical data such as color or other surface properties that are independent of the transformations between features are known as descriptors [13]. Descriptors are also used in feature-based registration methods, which primarily depend on unique, descriptive features in order to obtain a match between pointclouds [6]. These two types of data contained in a pointcloud are stored separately because they are different in nature and are processed differently in algorithms such as segmentation or registration.

1. **OVERVIEW OF APPROACH**

The proposed approach to automated weld path generation shown in figure 2 consists of a model preparation stage, a workspace sensing stage, a workpiece localization stage, followed by an offline robot path generation stage. The resulting path can be used to automate a welding process on the component in the workspace with a 6-DOF co-bot carrying a welding torch.

Uniqueness

* 1. **Model Preparation Stage**

In the model preparation stage, the geometry of the workspace and the workpiece is defined based on the prescribed application. An ideal model of the workpiece (including the weldment?) is generated using CAD. Part models are first generated of the individual workpiece components which are then assembled to represent the workpiece. The CAD assembly representing the workpiece is converted into a pointcloud through a uniform sampling technique to be used for workpiece registration. The pointcloud associated with the CAD model is known as the source pointcloud.

A simplified model of the workspace and environment including the welding table and the robot base is also created for simulation purposes. This environment model is also converted into a pointcloud file. The 3D models are generated using standard CAD software from which they can be exported as .ply files or other standard file formats.

* 1. **WORKSPACE SENSING STAGE**

Prior to the sensing stage, the workpiece is placed in the robot workspace by the operator in the proper relative orientation to be joined by a weldment. The relative orientation of the parts must match that of the model to an extent and the global location of the workpieces is restricted to the usable workspace of the robot.

In the sensing stage a sweeping motion of the arm is performed, and the workpiece and environment are scanned with the 2D LiDAR mounted on link 5 of the robot. Multiple 2D lidar scans are measured along with corresponding sensor poses. As the scanning stage continues, the data are transformed from the sensor frame link 5 to the base frame link 0 through the robot forward kinematics and accumulated into a 3D pointcloud with respect to the base frame. This process produces sparse data sets with redundant points. Therefore, the scans are filtered and downsampled using a voxel grid to improve results and reduce the resource requirements of storage and processing. The resulting pointcloud contains an image of the workpiece and fixtures as well as the top of the welding table and the background. The pointcloud associated with the LiDAR scan is known as the reference or target cloud. The sensing stage along with the methods of filtering and downsmpling can be seen below in figure 3.

* 1. **WORKSPACE LOCALIZATION STAGE**

In the workpiece localization stage, the source pointcloud derived from the CAD model is compared to the reduced reference cloud acquired from lidar in the sensing stage. The relative transformation between clouds is found using the iterative closest point algorithm (ICP). The pose of the workpeice can be used to determine the required location of the weld seam in a global sense.

The reference cloud, collected from LiDAR, contains a larger volume of points, but not necessarily more points, than the source cloud. Also, the percentage of the workpiece represented in the LiDAR cloud depends on the sweeping motion used in the scanning stage and the amount of interference caused by the clamps or other obstructions. In the best-case scenario, approximately half of the points associated with the external faces of the workpiece are available in the LiDAR cloud.

The LiDAR cloud is first reduced to the usable workspace of the robot using a 3D bounding box or PassThrough filter [] which removes points from the surrounding walls and extents of the table. Next, the point cloud is downsampled with a voxel filter [15] to ensure uniform density of points in the reference pointcloud and reduce computational requirements [] of the localization algorithms. The remaining image contains points from the workpiece, the clamps holding the workpiece, and the table. The robot arm may also be included in the remaining pointcloud. At this point, RANSAC based segmentation is used to compare geometrical information such as the planar nature of the table or the orthogonality of the workpiece to the LiDAR cloud to separate, or segment, the points associated with the workpiece. The results of a cascaded RANSAC segmentation are stored as the reference pointcloud cloud. Finally, the rigid transformation between the reference and source pointcloud is found with the iterative closest point (ICP) cloud registration algorithm. This transformation matrix represents the location and orientation of the workpiece with respect to a fixed origin.

* 1. **PATH GENERATION STAGE**

The weld seam is defined by the geometry of the workpiece and the desired weld. In the examples provided the seam is described as a collection of linear segments with defined torch angles. The required tool path which lies along the weld seam is transformed into global coordinates using the transformation resulting from the workpiece localization stage.

Model Preparation

Workpiece CAD

Model Generation



Conversion to

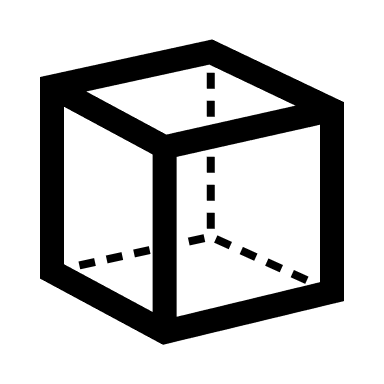
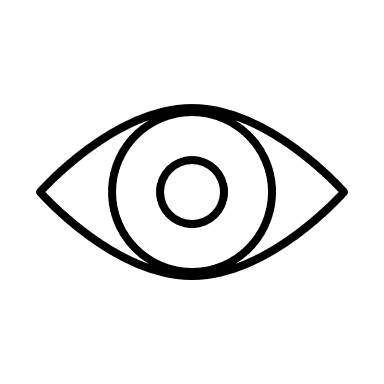
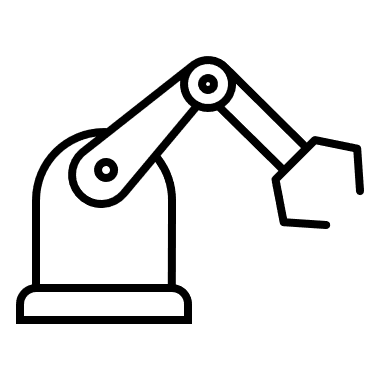
Pointcloud



Workspace Sensing

Collection of

2D LiDAR Scans



Conversion to

Pointcloud



Path Generation

Weld Seam

Transformation

Joint Velocity

Profile Generation

Workpiece Localization

PassThrough + Voxel

Filtering

RANSAC

Segmentation

ICP

Registration

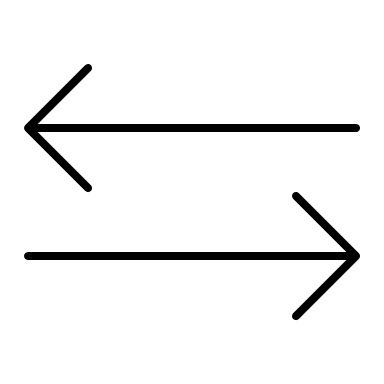
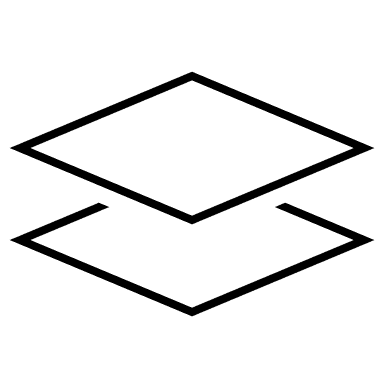


Figure 2 - Method for Automated Weld Path Generation

1. **Automated Weld Path Generation Approach**
   1. **Model Preparation Stage**

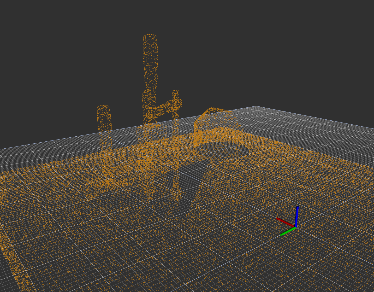


Fig 3b - After Voxel Filter

Fig 3c - After Bounding Box

Fig 3a - Before Filtering

Fig 3 - Filtering



**Filtering with PassThrough and Voxel**

The reference point cloud R generated from a LiDAR scan is defined as with equal to the number of data points in the original 3D scan, and the source pointcloud S generated from CAD is defined as . The cloud points are defined as triplets and with respect to a global origin fixed to the base of the robot.

Due to the nature of the LiDAR scanner, the reference cloud typically contains a large number of points not belonging to the workpiece. A PassThrough filter [33, PCL] removes points outside of prescribed limits which is used to eliminate data outside of the expected location of the workpiece. This reduces the computational requirements of the algorithms used in the following workpiece localization process by reducing the number of elements in the pointclouds.

The pointclouds S and R are restricted to the defined coordinate bounds resulting in a set of filtered clouds and .

The results of the PassThrough filter may still contain a large number of redundant points, and voxel grid down samplin is used to reduce the number of points in the reference and source clouds. The process is described in detail in [33,27 from 33] and shown below for convenience.

The coordinate system is divided into a uniform three-dimensional grid of cells referred to as voxels. The input cloud points are sorted into the grid by location, and then the centroid of the points in each voxel is added to the filtered point cloud [15]. This reduces redundant data points and allows for the resolution to be set using the voxel grid size as a parameter.

First, determine the maximum and minimum values in all three coordinates of the input cloud. Next calculate the cloud length width and height based on the maximum and minimum values.

Define the voxel grid unit side length , and let , , and represent the number of cells in each coordinate of the grid.

, ,

Next, sort each cloud point into the appropriate voxel.

Finally replace the cloud points in each voxel with the center of gravity of the voxel from points

* 1. **Segmentation with RANSAC**

Sample consensus represents a class of algorithms which can be used for fitting pointcloud data to various geometrical models such as lines, planes, and cylinders. The input points are sorted into inliers which fit the model and outliers which do not. Random sample consensus (RANSAC) involves repeated random sub-sampling to detect and segment pointcloud shapes [10][11][19]. This detection and segmentation of geometrical features in the pointcloud is useful in manufacturing applications such as welding in which the workpiece consists of well-defined geometrical shapes.

**Add math**

Although variations have been developed, the RANSAC algorithm is generally an iterative two-part process. The first part involves a hypothesis in which the first minimal sample set (MSSs) is selected at random from the input dataset (source pointcloud) then used as the basis for computing the model parameters. Following the selection of the minimal sample set, RANSAC checks which elements of the instantiated model are consistent with the entire dataset – these elements are referred to as the consensus set (CS) [10]. During each iteration, the instantiated model’s elements are compared to the original dataset. If the new MSS increases the number of correct correspondences compared to the best CS, it will then overwrite the previous CS. The algorithm continuously iterates and is terminated only when the CS reaches a certain threshold. When this threshold has been reached, the instantiated model based on model parameters of the newest MSS, the MSS is said to be consistent with the entire dataset.

Figure 4a – Input Cloud: Table, Workpiece, and Clamps

Figure 4 – Segmentation to Remove Table



Figure 4b – Plane Inliers: Table

Figure 4c – Plane Outliers: Workpiece and Clamps

* 1. **Iterative Closest Point**

The iterative closest point (ICP) algorithm is an optimization technique used to find a rigid transformation which best aligns the reference cloud with the source pointcloud set. This method considers the closest corresponding points between two pointclouds (point to point?) and estimates a transformation to minimize the distance between them using a method of least squares [6]. By iteratively registering the reference cloud with the source cloud and applying rotation matrix R and a translation vector *t,* the source point set is expected to converge as the correspondences achieve alignment*.* This method however, has also been proven to be locally convergent, which means that the algorithm easily fails when the rotation angle between two point-sets is large [7]. For this reason, a good initial transformation is important such that it guarantees that the algorithm converges to the global minimum [7]. In this work, false correspondences from the environment scans are known to inhibit convergence. An implementation to reduce these false correspondences, such as RANSAC, is included.

The primary challenge in the localization stage is the selection of point clouds to use as inputs to the ICP algorithm [4],[7]. It has been shown and verified in this work that the success of the alignment process is highly dependent on the correspondence between input data sets. The existence of points in one cloud which are not represented in the other cloud can only add cost [6] to the alignment process. Further, significant amounts of outliers will cause the alignment to fail or perform poorly. Modifications to ICP and alternative algorithms have shown improved performance [4] [6] [17] in the presence of outliers, and methods are available (used in this approach) for automatic rejection of non-corresponding outliers []. However, the approach in this work addresses the problem by reducing the reference point cloud to a subset of the LiDAR cloud which contains a portion of the workpiece without the surrounding table or clamps.

One of the key issues with standard ICP is that it may not reach the global minimum of convergence which can be due to things such as false correspondences. These false correspondences cause poor initial alignment and therefore increase the chance of getting stuck in a local minimum [7]. Outlier rejection based on Random Sample Consensus (RANSAC) is one of the several methods including distance-based rejection, or duplicate target point rejection, that reduces the number of outliers in pointclouds. Furthermore, RANSAC may also be utilized to provide a good initial guess for the transformation estimation in ICP [6].



Figure 6 – Workpiece Localization with ICP

“Classical ICP” uses the point-to-point error metric.

The ICP method as described in [12], [4] is summarized here. Let be the reference point cloud set for representing the workpiece to be aligned with the source point cloud set for where , and where each point corresponds to the point with the same index (*i=k*). The reference set will come from a CAD model while the source cloud set will be collected from a data scan of the real environment. The mean square objective function using the point-to-point error metric (or point-to-plane [4][6]) to be minimized is,

(1)

where is an array that projects onto and that translates onto . If the correct correspondences are known, the correct relative rotation/translation can be calculated in closed form. A closed form implementation that can be found in the PCL library [16] is briefly described. The center of mass for the reference point cloud set and source point cloud set respectively is calculated for each set as,

(2)

The reference point cloud set and source point cloud set are shifted by their center mass such that they are distributed around zero as,

(3)

(3)

A cross-covariance matrix, is defined as

(4)

A singular value decomposition (SVD) of is given as

(5)

where D is a diagonal matrix containing the singular values, , ordered such that and are the left and right singular vectors of . When the , the rotation and translation minimizing are unique and given by:

(6)

and

(7)

The correspondence between points assumed in equation (1) are unknown at the start. Therefore, this process is performed through iteration []. By assuming a correspondence between the reference and source point cloud set based on a minimum distance between points. The source is corrected and the process repeats until convergence of the source and reference point cloud set occurs as given in the error .

1. **PATH GENERATION**

**Transformation of Seam Points**

Consider a pointcloud set for located along the weld seam as described by the example application and defined in the CAD model. The weld seam exists at the shared location of the connecting faces of the two parts which make up the workpiece. The pointcloud set is a subset of the pointcloud set which represents the workpiece.

The pointcloud set and are defined with respect to a frame {{part}} located on the workpiece.

To complete the desired operation the robot must carry the torch along the weld seam. The path generation stage requires the robot tool path points to be described with respect to the fixed frame of the robot. The pointcloud sets and can be projected into the fixed frame of the robot with the rigid transformation for resulting from the ICP routine described previously as it is the best available approximation of the workpiece pose.

The set of points for is found by applying the rotation and translation separately as shown. = fn(R, t)

for

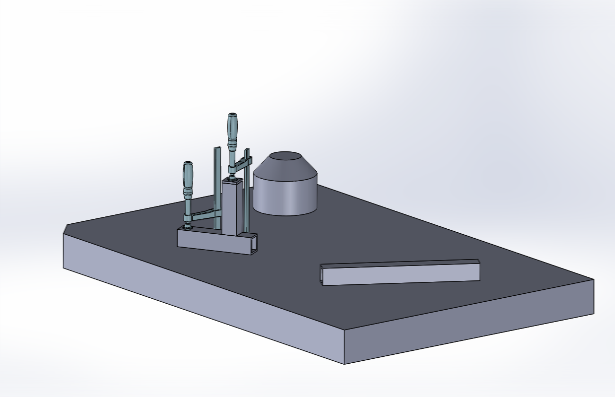


Figure 7 – Example Application A

**Joint Velocity Profile Generation**

The weld path is constructed from the information encoded in a solid model or similar representation of the manufactured part. This work will consider that the welds required for manufacturing are a collection of linear segments with defined work and travel angles for the torch, where the work angle is an angle measured about the weld seam from a reference plane and the travel angle is a rotation about an axis perpendicular to the weld seam and the torch axis. Finally, the torch can be rotated about the torch axis without affecting the weld so this angle (torch angle) can be freely chosen. The pose of the torch at any point will be contained in the pair of triplets (*Pi,Oi*) where *Pi* contains the *x,y,z* positions of the torch tip and *Oi* contains the three angles, work angle, travel angle and torch angle following a defined rotation operation (see the section, defining torch orientation). Thus, the weld path is first defined as a spline of linear segments defined by anchor or end points and orientations as pose pairs (*Pi, Oi*) existing at key points on the manufactured part (see Figure XXX Sample Manufactured part). The weld path then consists of intersecting line segments, for all weld paths *i=1…npath* with orientations *Oi,i+1* defined over each weld path between *Pi* and *Pi+1*. The trajectory is defined as a Linear segment with appropriate blends for the full robot pose. Parabolic blends are used in this work, but could be replaced with a blend that would limit the magnitude of higher derivatives of join motion. The weld specifications defines the travel speed along each line segment, and finally a nominal blend time for the end of the paths is selected as 2.5% of the total path time. The LSPB applied to each segment yields a function with *q* the function yielding robot configuration *C* at any time over the interval ti, *tf* the initial and final times determined from the linear segment speed and blend percentage, time. Note that required torch orientations are not continuous at the line intersection. In practice, one way of handling this discontinuity is by inserting an additional path segments between line segments , and , with end points with a small displacement along the along the weld seam toward point *i* or *i+2* and orientations at the endpoints, .

In order to complete the path generation, the key weld path information extracted from the sold model representation of the part is mapped to the reference frame using the transformation resulting from workpiece localization. The original weld path is derived from the key points and orientations in the manufactured part solid model: (*Pi,Oi*). The points are updated through multiplication with the transformation XXX. The orientations are updated from updates to the workpiece frame, .

Pi+1

Oi,i+1

P1+2

O1+1,i+2

Pi

Table X – Experimental Results – LiDAR Calibration



Figure XXX Sample Manufactured part

Defining the torch orientation: The fixed reference frame is while the workpiece frame is where lies along the weld seam and lies in the reference plane of the workpiece and is orthogonal to (see Figure YYY). Note that the weld seams are assumed linear and defined by end point pairs, . The rotation operator projecting the workpiece frame onto the fixed frame is

with a unit vector defined by the projection of onto frame {0}, a unit vector defined by the projection of onto frame {0} and completes the SO(3) operator. The torch frame orientation is aligned to the workpiece according to the so-called travel angle and work angle (and if desired the torch angle) which are called out in the weld specifications. The rotation operator projecting the torch frame onto the workpiece frame is consists of the three rotations, work angle (*w*) about , travel angle (*T*) about the new y axis and torch roll angle, (*R*) about the new x axis. The rotation operator projecting the torch frame onto the fixed frame is,

with

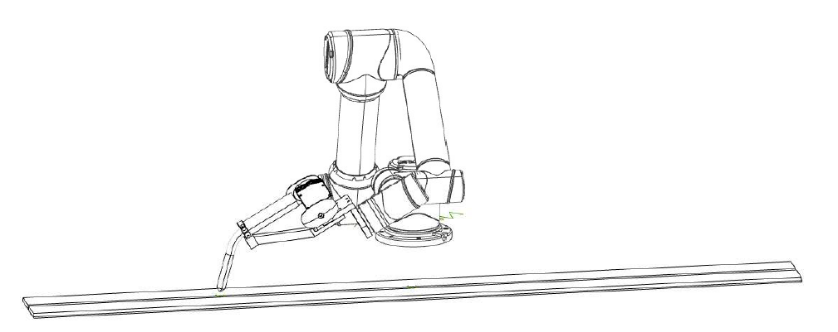
.

The orientation of the torch with respect to the fixed frame at position *i* will be represented as the triplet, using the definitions provide above.

Table Y – Experimental Results – Application A



*n*



*t*

*RP*

*xWP*

*w*

*x’*

*R*

*zWP, z’*

*y’, y’’*

*xT*

*w*

*T*

*zT*

*T*

*x0*

*y0*

*z0*

Figure YYY

1. **IMPLEMENTATION USING ROS AND PCL**

This research has been implemented in ROS on Ubuntu Linux which provides a multi-threaded and distributed software framework for robotics applications.

The combination of 2D LiDAR scans into 3D pointclouds was done using a custom ROS package *scan2cloud* that is based on *ROS* *laser\_geometry*. The rigid transformation from the sensor frame to the base of the robot is programmed with *ROS tf* so that individual 2D scans collected using *ROS rplidar* can be processed and saved as a .pcd file with respect to a global origin.



Figure 8 – Example Application B

The .pcd file is used for permanent storage and several pointcloud data types

The robot and sensor are operated simultaneously using ROS, and the resulting pointcloud is saved as a .pcd file which can be processed using the Point Cloud Library or converted to a .ply polygon file.

1. **MANUFACTURING APPLICATION**

A manufacturing task is considered, in which a weldment is performed on a workpiece resting on a welding table. The workpiece in this task consists of multiple components to be joined through weldment. The relative alignment of the multiple components of the workpiece is assumed to be correct within the physical constraints of the designed part prior to the automated process. In practice, this alignment is set by the operator and secured using clamps or other fixtures.

Variation in surface quality and workpiece dimension and shape are likely present however these are not the focus of this process. The workpiece geometries are generally assumed to match those in the model within a working tolerance. These local model inaccuracies certainly affect the global information produced regarding the geometry and location of the weld, but these affects are minor.

Two example applications are considered. In the first of which, two square tubes are joined perpendicular to one another with a fillet weld along two of the shared edges. In the second application, a square tube is joined to a flat plate with a fillet weld along the shared edge between the two components. In each of these examples, the assembly is temporarily joined together by clamps which will be included in the lidar scan. Prior to the alignment process, these clamps will be removed from the pointcloud data via segmentation with RANSAC.

In example application 1 the workpiece consists of two square tubes to be joined by weldment so that the tubes are perpendicular and form a tee.

In example application 2 the workpiece consists of a square tube to be joined by weldment to a flat plate so that the tube is perpendicular to the plate.

1. **SIMULATION RESULTS**

Example application A and B were performed for testing and validation of the proposed approach using synthetic data generated from CAD models of the workspace including the table and clamps as shown in the figure above. The models of the scene were converted to pointclouds using the uniform sampling process described for the conversion of the source cloud.



Simulation Results – Example Application A

Table Z – Simulation Results – Application A







Simulation Results – Example Application B



1. **EXPERIMENTAL RESULTS**

Example Application A was performed with an Aubo i5 on a welding table with a RP-LiDAR A2 mounted to the end effector for generating 3D pointsclouds shown in figure 9. In the scanning stage the arm performed a sweeping motion while collecting a point cloud containing approximately half of the workpiece, a large portion of the table, and a small portion of the arm itself. The recorded points are restricted to those that fall in a selected region of the usable workspace of the robot. This collection process produces redundant data points representing the objects and the cloud data can become large. The approach presented is applied to the raw projected LiDAR points as described with respect to the base frame of the robot.



Figure 9 - Experimental Setup for Application A

Calibration of 3D LiDAR

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |







Example Application A







1. **RESULTS AND DISCUSSION**

The results show that the proposed method of automated weld path generation effectively using RANSAC and ICP algorithms is viable for workpiece localization. The alignment achieved in the simulated examples is strong as shown in the table and proved that these algorithms can be implemented even in the case of increased noise in the target point cloud due to clamping mechanisms or miscellaneous items on the weld surface. The accuracy of the simulated tests is to be expected given that the simulated scene contained geometric features designed in CAD which closely resembled the target pointcloud features.

The physical experiment shows that this approach can be applied to a welding application using physical data in a realistic environment given that proper calibration is completed. The 3D LiDAR scans require calibration to accurately transform the source cloud to the target cloud. Further investigation is required to determine if the accuracy of the resulting workpiece localization is inside a working tolerance [24] for a welding operation.

The implementation of passthrough filtering voxel downsampling is an essential step prior to the segmentation algorithm with RANSAC. Furthermore, with the increased utilization of RANSAC, outlier rejection is a required step when there are features such as clamps included in the target pointlcoud set. Correspondence matching and alignment with ICP is proven to only be effective when sufficient outliers have been rejected and a good initial guess is provided by RANSAC. Utilization of cascaded RANSAC segmentation provides an effective means for sufficient outlier removal such that workpiece alignment accurate enough [24] to be applied to a physical welding process.

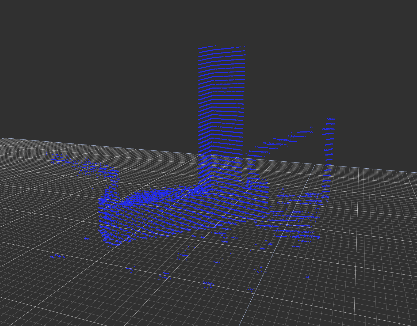
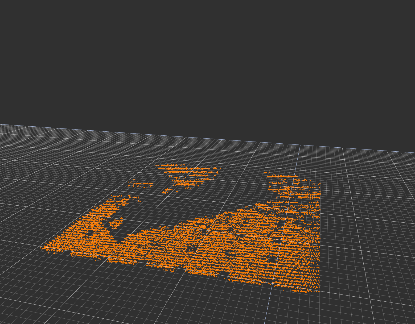


Figure 10 – Segmentation of Table

Figure 11 – Segmentation of Clamps

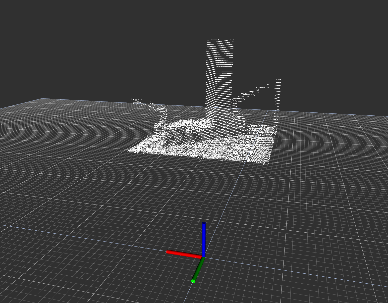
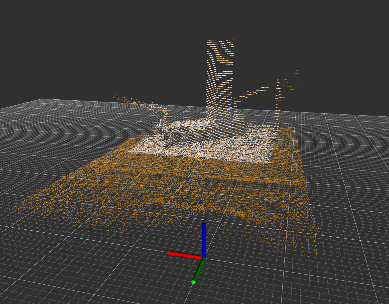
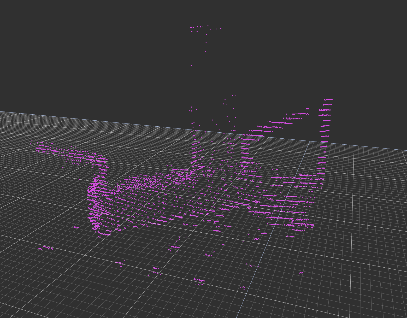
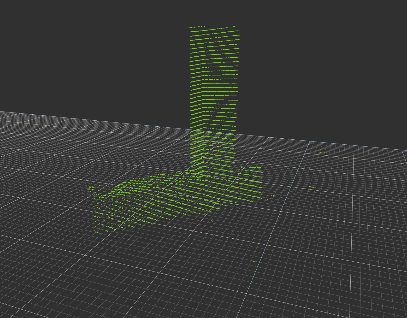
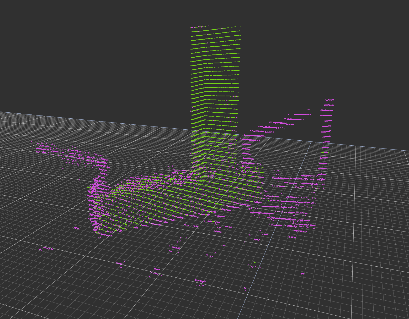


Figure 12 – Application A - Filtering

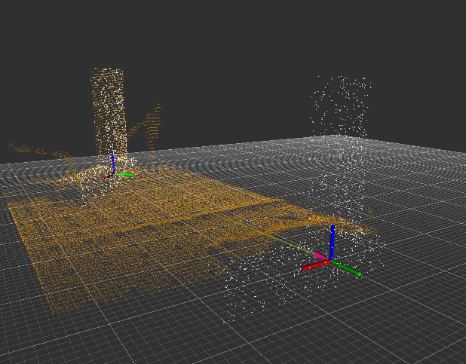
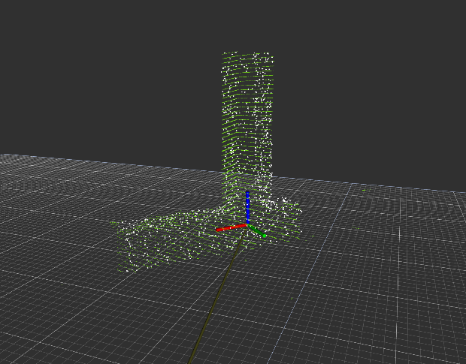


Figure 13 – Workpiece Localization with ICP for Example A

**REFERENCES**

[1] Segal, A. V., Haehnel, D., and Thrun, S., 2010, “Generalized-ICP,” Robotics: Science and Systems.

[2] Fischler, M. A., and Bolles, R. C., 1981, “Random Sample Consensus: A Paradigm for Model Fitting With,” Commun. ACM.

[4] Zhang, J., Yao, Y., & Deng, B. (2021). Fast and Robust Iterative Closest Point. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[5] Rusu, R. B., and Cousins, S., 2011, “3D Is Here: Point Cloud Library (PCL),” Proceedings - IEEE International Conference on Robotics and Automation.

[6] Zhao, H., Anwer, N., et. Al., 2016, “Registration with the Point Cloud Library PCL,” IEEE Int. Conf. Intell. Robot. Syst.

[7] Du, S., Xu, Y., Wan, T., Hu, H., Zhang, S., Xu, G., and Zhang, X., 2017, “Robust Iterative Closest Point Algorithm Based on Global Reference Point for Rotation Invariant Registration,” PLoS One.

[8] Schwarz, S., Preda, M., Baroncini, V., Budagavi, M., Cesar, P., Chou, P. A., Cohen, R. A., Krivokuca, M., Lasserre, S., Li, Z., Llach, J., Mammou, K., Mekuria, R., Nakagami, O., Siahaan, E., Tabatabai, A., Tourapis, A. M., and Zakharchenko, V., 2019, “Emerging MPEG Standards for Point Cloud Compression,” IEEE J. Emerg. Sel. Top. Circuits Syst.

[9] Galin, R., Meshcheryakov, R., Kamesheva, S., and Samoshina, A., 2020, “Cobots and the Benefits of Their Implementation in Intelligent Manufacturing,” *IOP Conference Series: Materials Science and Engineering*.

[10] Zuliani, M., 2008, “RANSAC for Dummies,” Citeseer.

[11] Li, L., Yang, F., Zhu, H., Li, D., Li, Y., and Tang, L., 2017, “An Improved RANSAC for 3D Point Cloud Plane Segmentation Based on Normal Distribution Transformation Cells,” Remote Sens.

[12] Besl, P. J., and McKay, N. D., 1992, “A Method for Registration of 3-D Shapes,” IEEE Trans. Pattern Anal. Mach. Intell.

[13] Wang, F., Liang, C., Ru, C., and Cheng, H., 2019, “An Improved Point Cloud Descriptor for Vision Based Robotic Grasping System,” Sensors (Switzerland).

[14] Moreno, C., and Li, M., 2016, “A Comparative Study of Filtering Methods for Point Clouds in Real-Time Video Streaming,” *Lecture Notes in Engineering and Computer Science*.

[15] Zygmunt, M., 2013, “The Testing of PCL: An Open-Source Library for Point Cloud Processing,” Geomatics, Landmanagement Landsc.

[16] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, May 9-13 2011.

[17] Efficient Variants of the ICP Algorithm (proper citation needed)

[18] Marani, A modified iterative closest point algorithm for 3D point cloud registration

[19] Schabel, Efficient RANSAC for Point-Cloud Shape Detection

[20] Schleth, Kuss, Kraus, Workpiece localization methods for robotic welding–a review

[21] Manufacturing Knowledge for Industrial Robot Systems:Review and Synthesis of Model Architecture

[22] Detection of workpiece shape deviations for tool path adaptation in roboticdeburring systems

[23] Weld Seam Detection using Computer Vision for Robotic Arc Welding

[24] Feature Extraction and Tracking of a Weld Joint for Adaptive Robotic Welding

[25] Chen, Robotic Welding Systems with Vision-Sensing andSelf-learning Neuron Control of Arc WeldingDynamic Process

[26] Teaser: Fast and Certifiable Point Cloud Registration

[27] Z. Gojcic, C. Zhou, J. D. Wegner, and A. Wieser, “The perfect match:3d point cloud matching with smoothed densities,” inProceedings ofthe IEEE Conference on Computer Vision and Pattern Recognition,2019, pp. 5545–5554.

[28] Y. Wang and J. M. Solomon, “Deep Closest Point: Learning Repre-sentations for Point Cloud Registration,” inIntl. Conf. on ComputerVision (ICCV), 2019.

[27] C. Choy, W. Dong, and V. Koltun, “Deep global registration,” inIEEEConf. on Computer Vision and Pattern Recognition (CVPR), 2020.

[28] Rajaraman, Dawson-Haggerty, Automated Workpiece Localization for Robotic Welding

[29] Kuss - Detection of Assembly Variations for Automatic Program Adaptationin Robotic Welding Systems

[30] Kuss - Automated planning of robotic MAG welding based on adaptive gap model

[31] Shah, Sulaiman, el al. , A review paper on Vision Based Identification, Detection and Tracking of Weld Seams Path in Welding Robot Environment

[32] Chen, The autonomous detection and guiding of start welding position for arc welding robot

[33] Fang, Chen, A Method of D-Type Weld Seam Extraction Based on Point Clouds , IEEE 2021

[34] Ji, Ren, An Improved Method for Registration of Pointcloud

[35] A vision based self tuning fuzzy controller for fillet weld seam tracking